

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS N/A		
2a. SECURITY CLASSIFICATION AUTHORITY N/A			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) N/A			5. MONITORING ORGANIZATION REPORT NUMBER(S) RADC-TR-88-175		
6a. NAME OF PERFORMING ORGANIZATION The University of Alabama in Huntsville		6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Rome Air Development Center (OCTS)		
6c. ADDRESS (City, State, and ZIP Code) Dept of Mathematics and Statistics Huntsville AL 35899			7b. ADDRESS (City, State, and ZIP Code) Griffiss AFB NY 13441-5700		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Rome Air Development Center		8b. OFFICE SYMBOL (If applicable) OCTS	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F49620-85-C-0013		
8c. ADDRESS (City, State, and ZIP Code) Griffiss AFB NY 13441-5700			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO. 62702F	PROJECT NO. 4506	TASK NO. 11
11. TITLE (Include Security Classification) IMPLEMENTATION OF ITERATIVE ALGORITHMS FOR AN OPTICAL SIGNAL PROCESSOR					
12. PERSONAL AUTHOR(S) Stephen T. Welstead					
13a. TYPE OF REPORT Final		13b. TIME COVERED FROM Mar 87 TO Mar 88		14. DATE OF REPORT (Year, Month, Day) August 1988	
15. PAGE COUNT 60					
16. SUPPLEMENTARY NOTATION N/A					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Signal Processing Noise Cancellers Optical Processing		
FIELD 09	GROUP 04	SUB-GROUP			
19. ABSTRACT (Continue on reverse if necessary and identify by block number) <p>The purpose of this study is to examine the possibility of implementing an iterative algorithm such as the conjugate gradient algorithm in an optical signal processor. This research is an extension of work done as part of the Summer Faculty Research Program (SFRP) in 1986 at RADC. The period of performance covered by this report is 1 March 1987 to 31 March 1988.</p> <p>The SFRP work focused on a prototype acousto-optic signal processor which was already in experimental operation as part of an RADC project (see (1,2)). This processor uses a variation of the Least Mean Square (LMS) algorithm. The goal of the current project is to investigate more powerful algorithms such as conjugate gradient that might provide improved performance for such a processor.</p>					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL Vincent C. Vannicola			22b. TELEPHONE (Include Area Code) (315) 330-4437		22c. OFFICE SYMBOL RADC (OCTS)

DD Form 1473, JUN 86

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

Contents

1. Introduction	1
2. Background on the Problem	1
2.1 The Signal Processing Application	1
2.2 The Least Mean Square Approach	2
2.3 Problems with LMS	3
3. Nonstationary Iterative Methods	7
3.1 New Approach to Iteration	7
3.2 Numerical Results	9
4. Analysis	13
4.1 Nonstationary Convergence Results	15
4.2 Error Analysis	16
5. Optical Systems	22
5.1 Hybrid System	22
5.2 All-Optical System	26
6. Concluding Remarks and Recommendations	28
6.1 The State of the Art	28
6.2 Recommendations	28
References	29
Publications and Presentations	30
Appendix 1 Preliminary Study of an Optical Implementation of the Conjugate Gradient Algorithm	

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

1. Introduction

The purpose of this study is to examine the possibility of implementing an iterative algorithm such as the conjugate gradient algorithm in an optical signal processor. This research is an extension of work done as part of the Summer Faculty Research Program (SFRP) in 1986 at RADC, Griffiss AFB, NY. The period of performance covered by this report is March 1, 1987 to March 31, 1988.

The SFRP work focused on a prototype acousto-optic signal processor which was already in experimental operation as part of an RADC project (see [1,2]). This processor uses a variation of the Least Mean Square (LMS) algorithm. The goal of the current project is to investigate more powerful algorithms such as conjugate gradient that might provide improved performance for such a processor.

2. Background on the Problem

2.1 The Signal Processing Application

The particular signal processing application is adaptive noise cancellation. A main signal is received consisting of the signal of interest $s(t)$ plus a noise signal $n(t)$. Omni-directional side antennas receive signals $n_j(t)$, $j=1,\dots,N$. A weighted combination of delayed versions of these side signals is used to estimate the noise $n(t)$. We denote this estimated noise by $y(t)$. The problem is to determine the optimum combination of weights in order to minimize the difference between the estimated noise and the actual noise.

The quantity we would like to minimize is

$$E(|e(t)|^2) \quad (2.1)$$

where $e(t)$, the so called 'error signal', is the difference between the main signal plus noise $s(t) + n(t)$ and the estimated noise $y(t)$, and E indicates expected value over all time with respect to some probability distribution. In practice, rather than a true expected value over all time, some finite measure or summation of recent signal history is used.

The expression (2.1) can be thought of as a functional (ie., real valued operator) of the unknown weight vector w used to form the estimated noise. It is well known [3] that the minimization of this functional is equivalent to setting its gradient equal to zero. This leads to the linear equation

$$Aw(x) = b(x) \quad (2.2)$$

where $w(x)$ is the unknown weight vector function evaluated at the delay point x , b is a vector function formed from the side signals and the main signal plus noise, and A is a positive definite symmetric operator corresponding to the covariance matrix in discrete formulations of this problem (see Appendix 1 or [4] for a discussion of the derivation of the analog version of this problem).

2.2 The Least Mean Square Approach

The formulation of the quantities A and b in equation (2.2) is a formidable computational task. As a result, several approaches have been advanced which attempt to circumvent this difficulty. One of these, the least mean square (LMS) algorithm (cf., [5]), has been implemented on several optical processors ([1], [6]), including the one under consideration here. It is the performance of this algorithm that we would like to improve upon.

Although the LMS algorithm is usually thought of as an approximation of more complicated algorithms for minimizing the quantity (2.1), one can also think of it directly as an algorithm for minimizing the quantity

$$|e(t)|^2 \quad (2.3)$$

instead of minimizing the quantity (2.1). As was the case before, this minimization problem is equivalent to setting a certain gradient equal to zero. In the case of a single side signal, the gradient associated with (2.3) is proportional to

$$e(t)n_1(t-x). \quad (2.4)$$

This gradient expression has the advantage of being easy to compute. In particular, it does not involve the calculation of a covariance matrix. However, the expression (2.3) only has a minimum in the case when (2.4) is zero. This can happen only when $e(t)$ is zero. But $e(t)$ has the form

$$s(t) + n(t) - y(t).$$

We hope to make the quantity $n(t) - y(t)$ zero, but in general $s(t)$ is not zero, and so $e(t)$ will also not be zero when there is a main signal present. This is a potential problem with LMS and we will consider it further in the next section.

Iterative processes have the general form

$$w_{i+1}(x) = w_i(x) + a_i p_i(x) \quad (2.5)$$

$$i = 0, 1, \dots$$

where $w_i(x)$ is the i^{th} iterative approximation of $w(x)$, $p_i(x)$ is a direction vector which indicates the direction to go in to get to the next iterate w_{i+1} , and a_i is the scalar stepsize that tells how far to go in the direction p_i .

For the LMS algorithm, we take $p_i(x)$ to be the vector given by (2.4) with $t = i\Delta t$, where Δt is the time increment between iterations. The stepsize is taken to be a sufficiently small fixed scalar a . As discussed in Appendix 1, it is possible to solve the LMS iteration process directly to obtain

$$w_k(x) = a \sum_{i=0}^{k-1} e_i n_1(i\Delta t - x), \quad (2.6)$$

where $e_i = e(i\Delta t)$. Letting $\Delta t \rightarrow 0$, we get the analog version of (2.6), namely

$$w(x) = a \int_0^t e(s) n_1(s-x) ds. \quad (2.7)$$

It is actually this solution, and not the iterative version of LMS, that is being implemented in the optical processors discussed in [1] and [6]. In this form, LMS is not a true iterative algorithm. Rather, it represents an approximate version of a complete solution of the minimization problem.

The advantage of LMS is the ease with which it can be implemented in a real time processor. The flow of data in such a processor is uninterrupted as the solution is continuously updated. This makes it particularly appealing for use in an optical processor. This is a desirable property of LMS that we should try to retain. Unfortunately, there are problems inherent in LMS that result in a degradation of performance that can reach unacceptable levels.

2.3 Problems with LMS

As mentioned in the previous section, there may be problems associated with LMS when a main signal is present (ie., signal-to-noise ratio (SNR) greater than 0). We can observe this phenomenon in the following numerical example (all numerical examples for this report were produced on a personal computer using Turbo-Pascal).

Figure 2.1 shows the performance of a numerical simulation of the LMS method in a case when the main signal $s(t)$ is 0. The signal received at the main antenna is just a noise signal $n(t)$ which we are attempting to cancel. In this example,

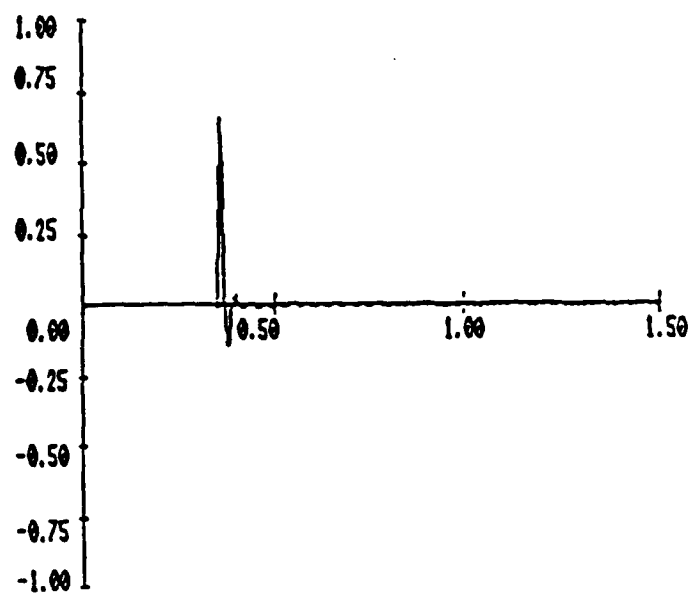


FIGURE 2.1 LMS WITH $SNR = 0$

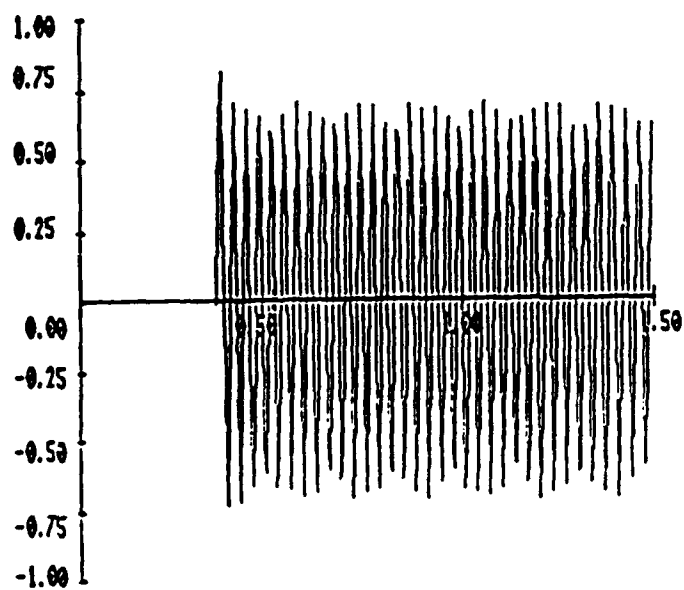


FIGURE 2.2 LMS WITH $SNR = 0.5$

$$n(t) = \sin(20\pi t). \quad (2.8)$$

The side antenna signal is of the form

$$n_1(t) = \sin(20\pi t + 0.1). \quad (2.9)$$

There are 30 delay taps spread over a delay aperture of 0.3 sec. The fixed stepsize is $a = 0.05$. As we can see in this figure, good noise cancellation is achieved after approximately half a second.

However, when even a small main signal is present, performance deteriorates drastically. Figure 2.2 shows the effect of adding a main signal of the form

$$s(t) = 0.5 \sin(30\pi t) \quad (2.10)$$

(so that the SNR is 0.5). The graph shows

$$n(t) - y(t) \quad (2.11)$$

the difference between actual noise and estimated noise. As one can see, there is essentially no noise cancellation. This is in agreement with observed experimental results [7], citing that LMS works well in "extremely poor SNR environments". Indeed, there is no hope of it working otherwise!

Why should this be the case? Recall that

$$e(t) = d(t) - y(t)$$

where

$$d(t) = s(t) + n(t)$$

is the signal received at the main antenna. If we substitute this expression for $e(t)$ in (2.4), and then use (2.4) as the direction vector p_i in (2.5), with $t = i\Delta t$, we obtain the following form for LMS:

$$w_{i+1}(x) = w_i(x) + a(d(i\Delta t) - y(i\Delta t))n_i(i\Delta t - x). \quad (2.12)$$

Convergence of this method implies

$$w_{i+1} \approx w_i$$

for large i , which, in turn, implies that the second term on the right side of (2.12) must converge to 0. But this implies

$$d(i\Delta t) - y(i\Delta t) \rightarrow 0$$

or, equivalently,

$$s(i\Delta t) + n(i\Delta t) - y(i\Delta t) \rightarrow 0.$$

But this quantity can never be 0 if $s(t)$ is independent (uncorrelated) of $n(t)$ and $y(t)$ (which we hope is the case if we are going to avoid cancelling the main signal!). Thus, LMS is trying to annihilate a quantity that can never be zero.

To put this another way, in the case when $s(t)$ is not identically zero, the quantity (2.3) has no minimum weight associated with it. LMS is trying to solve a problem that has no solution. The method which we introduce in the next section not only has better performance characteristics than LMS, but also completely avoids this serious drawback of LMS as a noise cancellation algorithm in the presence of a main signal.

3. Nonstationary Iterative Methods

3.1 New Approach to Iteration

We now consider a new way of incorporating iterative algorithms in a real time signal processing environment. The motivation for the approach is optical signal processing, which allows us the computational speed to consider such an approach. The uniqueness of the method lies in the fact that the data flow is allowed to drive the iterations, providing effective real time performance. Rather than perform multiple iterations on a fixed problem, which must be formulated from stored data, we allow variations in the incoming data to continuously update the problem while iterations are being performed. This is well suited to optical processing, where data storage and retrieval can be a problem, but computational speed is not. The result is an adaptive process that can significantly outperform the traditional LMS algorithm.

In contrast to the LMS algorithm, the new iterative technique deals with equation (2.2) directly, rather than an approximation of that equation. To illustrate the technique, we consider the simplest type of iterative algorithm of the form (2.5), namely the steepest descent algorithm with fixed stepsize. This algorithm has the form

$$\begin{aligned}w_{n+1} &= w_n + a r_n \\ r_n &= b - A w_n.\end{aligned}\tag{3.1}$$

The fixed scalar a is the stepsize. The sequence $\{w_n\}$ constructed from (3.1) will converge to the solution w^* of (2.2) provided

$$a < 1/M$$

where M is the largest eigenvalue of A (cf. [8]).

The usual approach in implementing an algorithm such as (3.1) is to compute A and b from the input data once, and then to regard them as fixed while the iterations are being performed. However, for our real time acousto-optic processor, it is easier to recompute A and b on every iteration, rather than to store and retrieve their values. This recomputation of A and b , however, introduces variations in their values as the iterations are being performed. Thus, it is more appropriate to write the algorithm (3.1) in the form

$$\begin{aligned}w_{n+1} &= w_n + a r_n \\ r_n &= b_n - A_n w_n\end{aligned}\tag{3.2}$$

where A_n and b_n are the updated versions of A and b at the n^{th} iteration.

The algorithm (3.2) is an example of a nonstationary iterative process as defined for example in [9]. In practice, one finds that A_n and b_n do in fact change on every iteration. What remains the same, however, is that the sequence of problems

$$A_n w = b_n, \quad n = 0, 1, 2, \dots \quad (3.3)$$

all have the same solution w^* for each value of n (or, at least, w^* changes slowly in time compared to the speed of the iteration process).

This makes sense in the context of our noise cancellation problem. Recall that the weight vector solution w^* represents which of the delayed versions of the side signal are to be weighted. This is not going to change from one iteration to the next. Thus, the solution does not change, even though the formulated problem changes from one iteration to the next.

When the solution does change over time, this type of process will adapt to the new solution since we are always incorporating the most recent signal data. Moreover, convergence to the new solution value should be very quick since the old solution value provides a good starting point from which the iteration process can seek the new solution.

Other iterative algorithms can also be put in nonstationary form. One improvement on the steepest descent algorithm is to optimize the stepsize at each iteration step. The nonstationary version of this algorithm has the form

$$\begin{aligned} w_{n+1} &= w_n + a_n r_n \\ r_n &= b_n - A_n w_n \\ a_n &= (r_n, r_n) / (r_n, A_n r_n). \end{aligned} \quad (3.4)$$

The nonstationary conjugate gradient algorithm takes the form

$$\begin{aligned}
w_{n+1} &= w_n + a_n p_n \\
p_{n+1} &= r_{n+1} - c_n p_n \\
a_n &= (r_n, p_n) / (p_n, A_n p_n) \\
c_n &= (r_{n+1}, A_n p_n) / (p_n, A_n p_n) \\
r_n &= b_n - A_n w_n.
\end{aligned} \tag{3.5}$$

Here, a_n and c_n are scalars, (\cdot, \cdot) indicates inner product, and p_n is the direction vector. In the next section, we show numerically that sequences $\{w_n\}$ generated from either (3.2), (3.4) or (3.5) will converge to the common solution w^* of the sequence of problems (3.3). In section 4.1 we look at analytical results concerning the convergence of such sequences to the desired solution w^* .

3.2 Numerical Results

This section presents the results of three numerical simulations comparing the performance of several nonstationary iterative algorithms and the traditional LMS algorithm. As mentioned previously, all numerical results were produced on a personal computer. In order to be computationally feasible on such a computer, the examples are constructed so that an exact solution is possible with a relatively small number of tap weights (we choose 6 tap weights for the iterative algorithms and 30 for LMS). In order to study the behavior and stability of the methods for larger numbers of tap weights, more computer power will be needed. For an optical processor, however, large numbers of tap weights will present no computational difficulty.

EXAMPLE 1: For the first example, we have no main signal, so that $\text{SNR} = 0$. The noise signal to be cancelled is

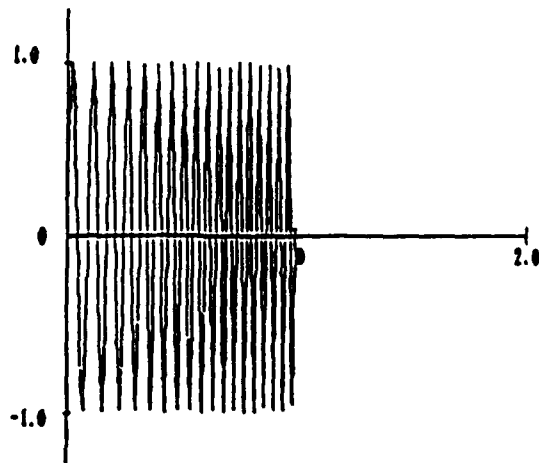
$$n(t) = \sin(20\pi t + 50t^2), \quad 0 < t < 1.$$

The graph of $n(t)$ is shown in Figure 3.1 (a). A single side antenna receives a copy of the noise signal in the form

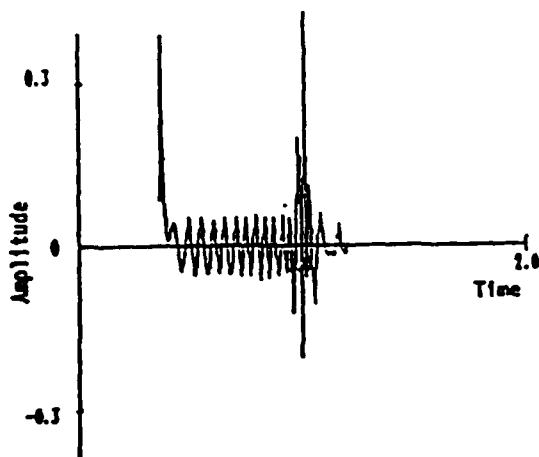
$$n_1(t) = n(t + 0.1).$$

Delayed versions of this side antenna signal are formed over a total delay aperture of $R = 0.3$.

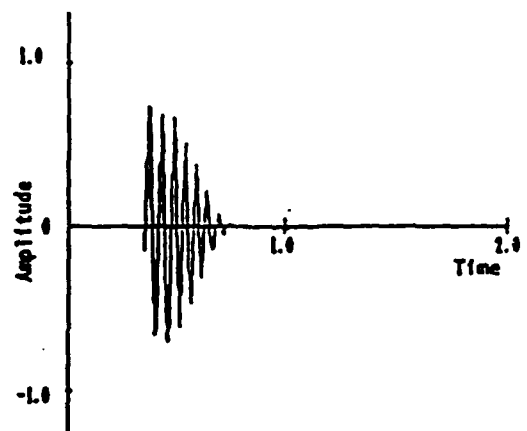
Figure 3.1 (b) shows the results for the LMS algorithm. The algorithm is run with a fixed stepsize of 0.1 and 30 delay taps. 200 iterations are used over a time interval from $t = 0.35$ to $t = 1.3$ (ie., at each iteration the current time is updated by a time increment of $\Delta t = (1.3 - 0.35)/200$). The graph



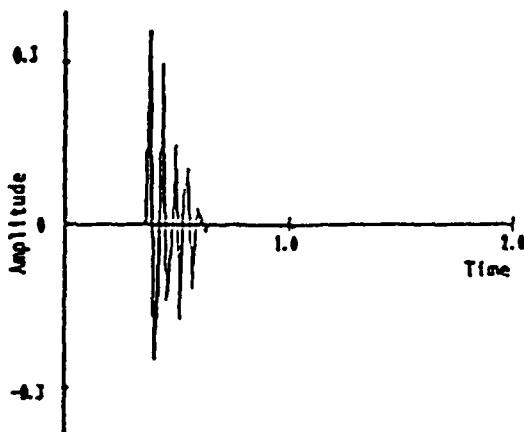
(a) NOISE SIGNAL



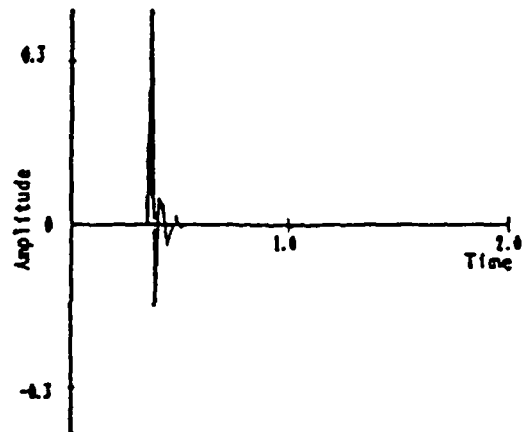
(b) LMS FILTER OUTPUT



(c) NONSTATIONARY STEEP. DESC.



(d) NONSTATIONARY OPT. STEEP. DESC.



(e) NONSTATIONARY CONJ. GRAD.

FIGURE 3.1 EXAMPLE 1

shows the difference between actual noise and estimated noise. We observe that this output noise settles down to a signal of amplitude 0.05, although the algorithm does display some problems near $t=1$, where the noise signal becomes compressed (higher frequency).

Figures 3.1(c)-(e) show numerical results for, respectively, the nonstationary steepest descent, with fixed and optimized stepsize, algorithm and conjugate gradient algorithm. The number of delay taps used is 6, so that the covariance matrices A_n are 6×6 , and the vectors b_n have 6 components. The entries in A_n and b_n are, respectively, auto-correlation and cross-correlation functions, which are computed using integration over time. Theoretically, this integration should be performed over the time interval $-\infty$ to ∞ . However, in practice this integration can only be done over a finite interval. We choose the interval from $t_0 - 3$ to t_0 , where t_0 is current time. The integration is performed numerically in the simulations using a 200 point Simpson's rule.

For these nonstationary algorithms, the values of A_n and b_n are recomputed on every iteration. The simulations are run from time $t = 0.35$ to $t = 1.3$. At each iteration, the current time is updated by an amount $\Delta t = (1.3-0.35)/(\# \text{ iterations})$.

From Figures 3.1(c)-(e), one can see that in this example the nonstationary iterative algorithms provide a significant improvement in performance over the LMS algorithm. The complexity of the noise signal causes no difficulties for these algorithms. Not surprisingly, the best performance is obtained from the conjugate gradient algorithm, computationally the most complex of the algorithms.

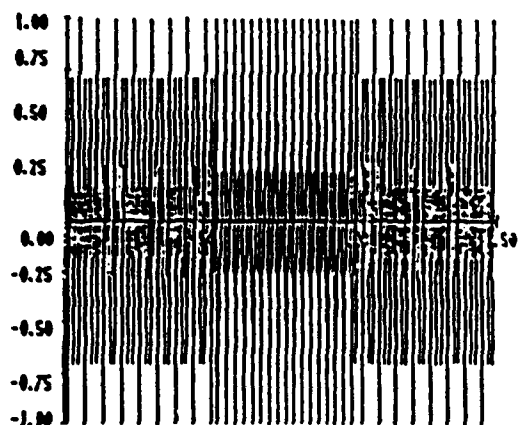
EXAMPLE 2: This is another example with $\text{SNR} = 0$. We consider a noise signal, shown in Figure 3.2(a), of the form

$$n(t) = \begin{cases} \sin(50\pi t) & 0 < t < 0.5 \\ \sin(100\pi t) & 0.5 < t < 1.0 \\ \sin(50\pi t) & 1.0 < t < 1.5. \end{cases}$$

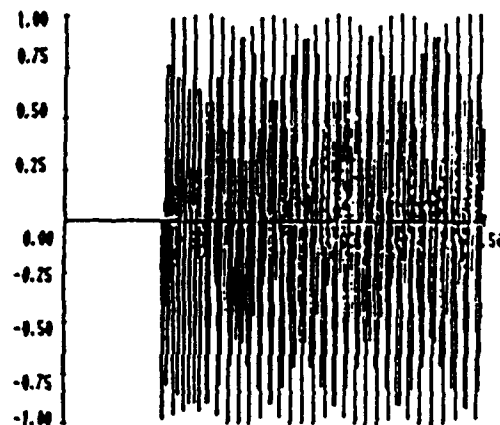
A side antenna receives a signal of the form

$$n_1(t) = n(t + 0.1).$$

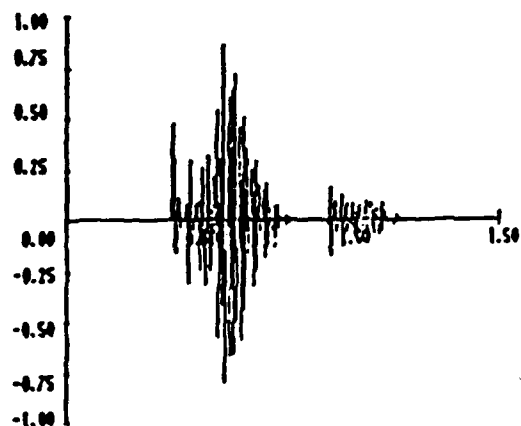
This particular noise signal was chosen to provide another example where the LMS algorithm has apparent difficulty.



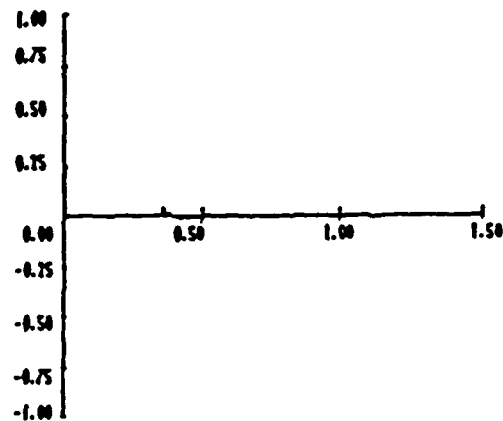
(a) NOISE SIGNAL



(b) LMS FILTER OUTPUT



(c) NONSTATIONARY STEEP. DESC.



(d) NONSTATIONARY CONJ. GRAD.

FIGURE 3.2 EXAMPLE 2

The LMS algorithm was run with a stepsize of 0.000001, with all other parameters being the same as in the previous example. Figure 3.2(b) shows the output of this algorithm. As one can see, there is essentially no noise cancellation. Larger numbers of iterations, and larger and smaller stepsizes produced no better results.

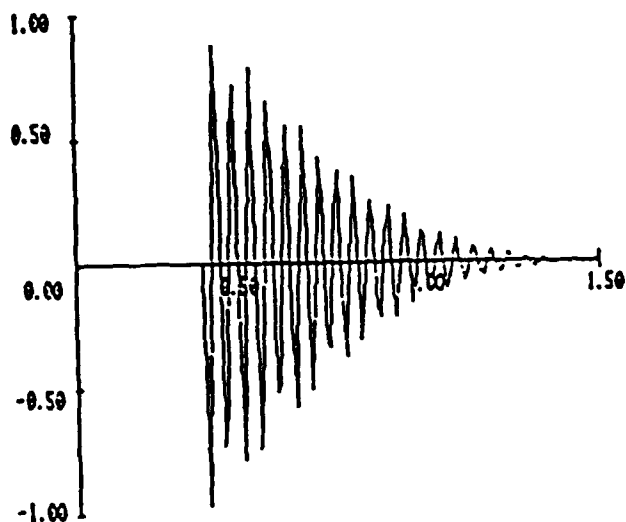
Figures 3.2(c)-(d) show the results for nonstationary steepest descent and nonstationary conjugate gradient algorithms. The noise cancellation is similar to the previous example, and is much better than LMS.

EXAMPLE 3: For our final example, we revisit the problem considered in Section 2.3. Recall that the LMS algorithm did not work at all in the presence of a main signal. Figures 3.3 (a)-(b) show the results of applying the nonstationary steepest descent with fixed stepsize and nonstationary conjugate gradient algorithms to the same problem. The noise signal is defined by (2.8), with side signal given by (2.9) and main signal given by (2.10). As one can see from the figures, the performance of these algorithms is not affected by the presence of a main signal. Figures 3.3(c)-(d) show the effect of an even larger SNR of 10. The steepest descent algorithm remains unaffected, while there is some deterioration in the performance of the conjugate gradient algorithm. It is believed that this is due to the effect of the large magnitude of $s(t)$ on the numerical integration scheme, and not due to the conjugate gradient algorithm itself. In this example, apparently conjugate gradient is more sensitive than steepest descent to errors in the computation of A_n and b_n . This is not believed to generally be the case.

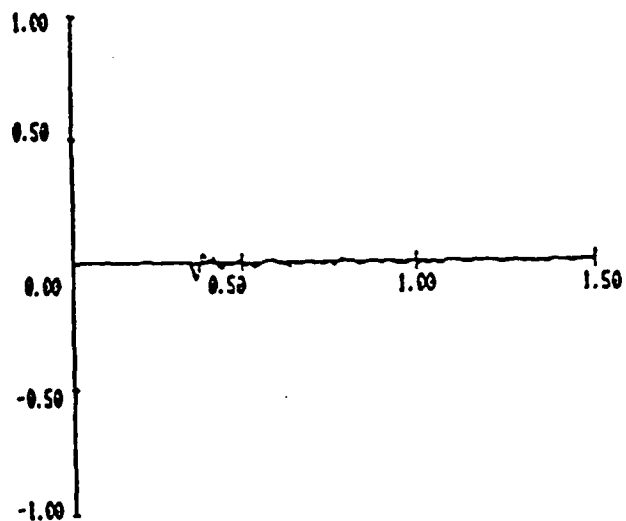
What these examples show is that there are situations where LMS does not work at all as a noise cancellation algorithm. We have shown that nonstationary iterative algorithms will work in these same situations. Since these simulations were run on a PC, the examples had to be set up so that a solution could be attained with a small number of tap weights (6). The performance of these nonstationary algorithms should be investigated on larger computers using a greater number of tap weights. Matrix pre-conditioning techniques may be necessary in this case to deal with possible ill conditioning effects.

4. Analysis

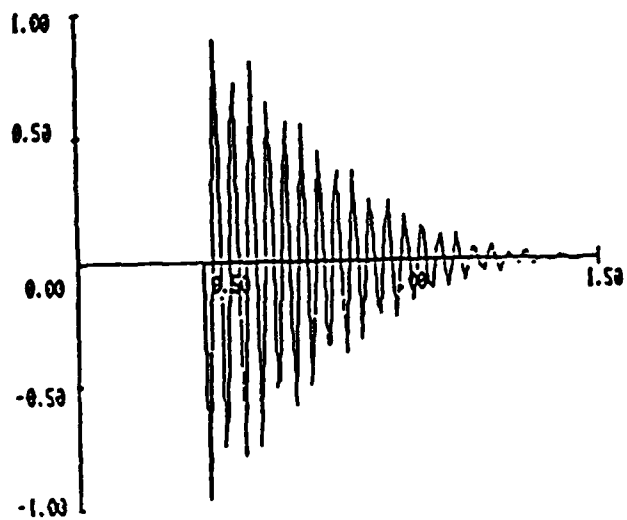
Not much is available in the literature concerning analysis results for nonstationary iterative processes of the type we are considering here. This is not surprising, since, without optical processing, such a process presents a formidable computational task. The next section contains a convergence proof for the nonstationary steepest descent algorithm. In Section 4.2, convergence results are combined with



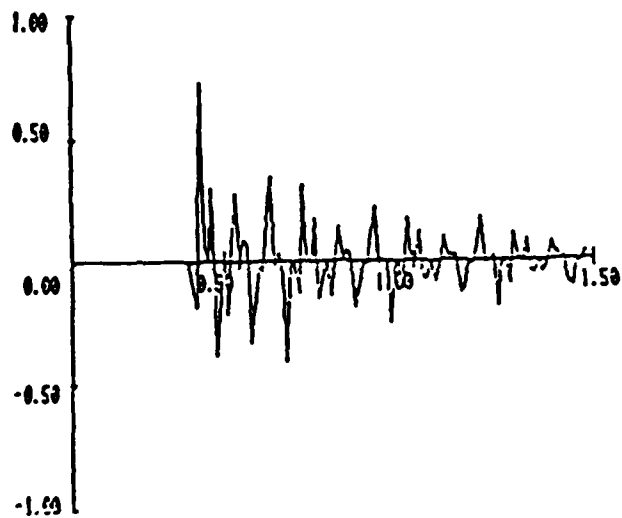
(a) NONSTATIONARY STEEP. DESC.
WITH SNR = 0.5



(b) NONSTATIONARY CONJ. GRAD.
WITH SNR = 0.5



(c) NONSTATIONARY STEEP. DESC.
WITH SNR = 10



(d) NONSTATIONARY CONJ. GRAD.
WITH SNR = 10

FIGURE 3.3 EXAMPLE 3

perturbation results to produce an error analysis for this algorithm.

4.1 Nonstationary Convergence Results

The situation we are considering is as follows. We have a sequence $\{A_k\}$ of positive definite symmetric linear operators (for example, covariance matrices) and a sequence $\{b_k\}$ of vectors such that the equations

$$A_k w = b_k, \quad k = 0, 1, 2, \dots \quad (4.1)$$

have a common solution w^* . Let the scalar a be such that

$$\|I - a A_k\| \leq \xi < 1 \quad (4.2)$$

for each $k = 0, 1, 2, \dots$, and some $\xi < 1$. The operator I is the identity operator. This is not an unreasonable condition since a similar condition is necessary for convergence of the normal steepest descent process [10]. We then have that the sequence $\{w_k\}$ generated by the process

$$w_{k+1} = w_k + a(b_k - A_k w_k), \quad k = 0, 1, 2, \dots \quad (4.3)$$

converges in norm to w^* .

To prove this, note in the following that

$$b_k - A_k w^* = 0$$

so that we have

$$\begin{aligned} \|w_{k+1} - w^*\| &= \|w_k + a(b_k - A_k w_k) - w^*\| \\ &= \|w_k - w^* + a(b_k - A_k w_k) - a(b_k - A_k w^*)\| \\ &= \|w_k - w^* - aA_k(w_k - w^*)\| \\ &= \|(I - aA_k)(w_k - w^*)\| \\ &\leq \|I - aA_k\| \|w_k - w^*\| \end{aligned}$$

$$\begin{aligned} &\leq \xi \|w_k - w^*\| \\ &\leq \xi^{k+1} \|w_0 - w^*\|. \end{aligned}$$

Since $\xi < 1$, this last term $\rightarrow 0$ as $k \rightarrow \infty$. This completes the proof.

As a corollary, we note that it suffices to replace condition (4.2) with

$$\|I - aA_k\| = \xi_k < 1 \quad (4.4)$$

for each k . We then find that

$$\|w_{k+1} - w^*\| \leq \left(\prod_{j=0}^k \xi_j \right) \|w_0 - w^*\|$$

and the term on the right side also $\rightarrow 0$ as $k \rightarrow \infty$ since each factor in the product is < 1 . A sufficient condition for satisfying (4.4) is

$$a < m_k \quad (4.5)$$

where m_k is the smallest eigenvalue of A_k .

In [11], convergence results are given for the case when the sequence of operators $\{A_k\}$ satisfies $A_k \rightarrow A$ for some fixed operator A . However, the situation considered here, namely that the equations (4.1) have a common solution, seems to better reflect what would happen in practice. Table 4.1 shows data taken at three different time steps during one of the simulation runs discussed in the previous section. The three matrices shown here are obviously very different. What is the same is the solution $w = (0,0,1,0,0,0)$ to the three linear equations represented by these matrices and vectors.

4.2 Error Analysis

In [12], a nonstationary perturbation analysis is given for the stationary steepest descent algorithm. That is, the fixed problem

$$Aw = b$$

is solved using the usual steepest descent algorithm, and the effects of different perturbations

Covariance Matrix 1:

0.100725	-0.003987	-0.018533	-0.002036	-0.011653	0.003697
-0.003987	0.092327	-0.012011	-0.023480	-0.000029	-0.003673
-0.018533	-0.012011	0.083455	-0.019583	-0.024551	0.007741
-0.002036	-0.023480	-0.019583	0.074181	-0.024853	-0.019566
-0.011653	-0.000029	-0.024551	-0.024853	0.066859	-0.026961
0.003697	-0.003673	0.007741	-0.019566	-0.026961	0.058625

b-Vector 1:

-0.01853 -0.01201 0.08345 -0.01958 -0.02455 0.00774

Solution 1:

-0.00000 0.00000 1.00000 0.00000 0.00000 0.00000

Covariance Matrix 2:

0.133363	-0.125000	0.116637	-0.108363	0.100000	-0.091637
-0.125000	0.125000	-0.116637	0.108363	-0.100000	0.091637
0.116637	-0.116637	0.116637	-0.108363	0.100000	-0.091637
-0.108363	0.108363	-0.108363	0.108363	-0.100000	0.091637
0.100000	-0.100000	0.100000	-0.100000	0.100000	-0.091637
-0.091637	0.091637	-0.091637	0.091637	-0.091637	0.091637

b-Vector 2:

0.11664 -0.11664 0.11664 -0.10836 0.10000 -0.09164

Solution 2:

-0.00000 0.00000 1.00000 -0.00000 0.00000 -0.00000

Covariance Matrix 3:

0.167037	0.037466	-0.009107	-0.002472	-0.013037	-0.000387
0.037466	0.157761	0.036683	-0.000259	0.003787	-0.013928
-0.009107	0.036683	0.150458	0.036188	0.006093	0.010720
-0.002472	-0.000259	0.036188	0.141609	0.031116	0.008366
-0.013037	0.003787	0.006093	0.031116	0.132684	0.026026
-0.000387	-0.013928	0.010720	0.008366	0.026026	0.125200

b-Vector 3:

-0.00911 0.03668 0.15046 0.03619 0.00609 0.01072

Solution 3:

-0.00000 -0.00000 1.00000 0.00000 -0.00000 0.00000

TABLE 4.1 THREE DIFFERENT MATRIX PROBLEMS WITH SAME SOLUTION

introduced at each step of the iteration process are studied. Thus, at the n^{th} step, instead of having exactly A and b available, we assume that we are dealing with perturbed versions of these quantities:

$$\begin{aligned} A + \delta A_n \\ b + \delta b_n. \end{aligned}$$

The analysis provides a bound for the difference between the perturbed iterates \bar{w}_n and the normal unperturbed iterates w_n .

In this section we apply these ideas to the nonstationary steepest descent process. As before, we consider a sequence of problems

$$A_n w = b_n, \quad n = 0, 1, 2, \dots \quad (4.6)$$

with common solution w_* . We now introduce perturbations δA_n and δb_n at each iteration step, so that we obtain a sequence of perturbed problems of the form

$$\bar{A}_n w = \bar{b}_n$$

where

$$\begin{aligned} \bar{A}_n &= A_n + \delta A_n \\ \bar{b}_n &= b_n + \delta b_n. \end{aligned}$$

This is a particularly important problem to consider in the context of optical implementation, since we can expect errors in the formulation of A and b at each iteration step. We now determine the effect of these errors.

The nonstationary steepest descent algorithm applied to the sequence of problems (4.6) has the form

$$\bar{w}_{n+1} = \bar{w}_n + a(\bar{b}_n - \bar{A}_n \bar{w}_n), \quad n = 0, 1, 2, \dots \quad (4.7)$$

We assume that the stepsize a has been chosen to satisfy the condition (4.2). The nonstationary process generates a sequence $\{\bar{w}_n\}$. From a practical point of view, what we would like to know is: for large n , how far off is the perturbed iterate \bar{w}_n from the true solution w_* of the unperturbed system (4.1)?

We answer this question in several stages. First, we determine the maximum difference between the perturbed iterate \tilde{w}_n and the corresponding iterate w_n from the unperturbed process (4.3). The analysis here is very similar to that given in [12], so we only sketch the details.

Define

$$\delta w_n = \tilde{w}_n - w_n, \quad n = 0, 1, 2, \dots$$

Subtracting equation (4.3) from (4.7), we find that δw_n satisfies a nonhomogeneous difference equation of the form

$$\delta w_{n+1} = (I - a\tilde{A}_n)\delta w_n + a g_n, \quad n = 0, 1, 2, \dots \quad (4.8)$$

where

$$g_n = \delta b_n - \delta A_n w_n.$$

Equation (4.8) can be solved directly to obtain

$$\delta w_{n+1} = \sum_{k=0}^n \left\{ \prod_{j=k+1}^n (I - a\tilde{A}_j) \right\} a g_k$$

(see [12] for the precise meaning of the noncommutative product of operators on the right). Thus,

$$\|\delta w_{n+1}\| \leq a \left(\sum_{k=0}^n \left\{ \prod_{j=k+1}^n \|I - a\tilde{A}_j\| \right\} \right) \left(\max_{k \leq n} \|g_k\| \right). \quad (4.9)$$

Denote

$$\alpha \equiv \sup_n \|\delta A_n\|$$

$$\beta \equiv \sup_n \|\delta b_n\|$$

$$W \equiv \sup_n \|w_n\|.$$

α and β are finite by assumption and W is finite since we assume $\{w_n\}$ is a convergent sequence. Then

$$\max_{k \leq n} \|g_k\| \leq \beta + \alpha W.$$

Also,

$$\begin{aligned} \|I - a\bar{A}_j\| &\leq \|I - aA_j\| + a\alpha \\ &< \xi + a\alpha \end{aligned}$$

so that

$$\sum_{k=0}^n \left\{ \prod_{j=k+1}^n \|I - aA_j\| \right\} \leq \sum_{k=0}^n \left\{ \xi + a\alpha \right\}^k. \quad (4.10)$$

The right side of (4.10) converges to

$$\frac{1}{1 - (\xi + a\alpha)}$$

as $n \rightarrow \infty$, provided

$$\xi + a\alpha < 1. \quad (4.11)$$

Thus, in this case we have from (4.9),

$$\sup_n \|\delta w_n\| \leq \frac{a}{1 - (\xi + a\alpha)} (\beta + \alpha W). \quad (4.12)$$

If we define m_j as the smallest eigenvalue of A_j and let m equal the infimum of the sequence $\{m_j\}$ then

$$\|I - aA_j\| = 1 - am_j \leq 1 - am.$$

If we assume $m > 0$ and take

$$\xi = 1 - am < 1$$

the condition (4.11) becomes

$$\alpha < m$$

and the bound (4.12) can be written as

$$\sup_n \|\bar{w}_n - w_n\| \leq \frac{1}{m - \alpha} (\beta + \alpha W). \quad (4.13)$$

This implies that the perturbed process (4.7) could become unstable if the perturbations on A_n exceed m .

We can now estimate the difference between \tilde{w}_n and w^* :

$$\| \tilde{w}_n - w^* \| \leq \| \tilde{w}_n - w_n \| + \| w_n - w^* \|.$$

For sufficiently large n , given $\epsilon > 0$ we can write

$$\| \tilde{w}_n - w^* \| \leq \frac{1}{m - \alpha} \left(\beta + \alpha W \right) + \epsilon. \quad (4.14)$$

This is the desired result providing the distance of the perturbed iterates \tilde{w}_n from the true solution w^* . Not surprisingly, this distance depends on the size of the errors α and β , and this distance can blow up very quickly if α is close to m .

For the case of a single fixed equation, error bounds analogous to (4.14) are frequently written in terms of the condition number of a matrix. We can obtain a similar result here, if we agree to define the "condition number of the sequence $\{A_n\}$ " to be the quantity

$$\Gamma \equiv \frac{M}{m}$$

where M is the supremum of the sequence $\{M_n\}$, where M_n is the maximum eigenvalue of A_n . We assume M is finite, as well as the quantity

$$B \equiv \sup_n \| b_n \|.$$

Then from (4.14) we get a bound for the relative error:

$$\| \tilde{w}_n - w^* \| / W \leq \frac{\Gamma}{1 - \left(\frac{\alpha}{m} \right)} \left(\frac{\beta}{B} + \frac{\alpha}{M} \right).$$

Thus the relative error in \tilde{w}_n is proportional to the relative errors in $\{\tilde{A}_n\}$ and $\{\tilde{b}_n\}$. The constant of proportionality is dependent on Γ , the condition number of the sequence $\{A_n\}$, as well as the proximity of α to m . Once again we observe that the process can become unstable if the perturbations (α) on A_n exceed the smallest (m) of the eigenvalues of all the A_n .

5. Optical Systems

In this section we consider two approaches for possible optical implementation of the nonstationary iterative algorithms discussed in the previous sections. The first of these is a hybrid system that would use optics to do the bulk of the computational effort and an electronic microprocessor to perform the actual algorithm iteration step. This approach allows some flexibility in the choice of the algorithm, although its performance would be limited by the optics to electronics conversion. The second approach is an all optical implementation of the nonstationary steepest descent with fixed stepsize algorithm. This processor would be able to run just the one algorithm. It would, however, be an important step toward realizing all-optical implementations of the other algorithms, such as conjugate gradient, and it would provide real time performance.

5.1 Hybrid System

The first approach we consider is an electro-optic hybrid system. This system will use optics to do the hard computational task of computing the covariance matrix A_n and the vector b_n on every iteration. These computations involve correlations and integrations which can be easily accomplished optically. The iteration step of algorithms such as (3.4) and (3.5), however, involve scalar division which cannot easily be done in the optics domain. An electronic microprocessor will be used to perform this step. The use of a programmable microprocessor here will also allow the testing and comparison of different algorithms in a real signal environment. The division of tasks between optics and electronics in this hybrid processor is shown in Figure 5.1.

An overview of the hybrid system is shown in Figure 5.2. A single side signal $n_1(t)$ will pass through a tapped delay line and drive an array of light emitting diodes (LED's). The LED's are a low cost alternative to a laser system. Also, unlike lasers, the LED's have linear characteristics over a broad range in converting the input electrical signal into light, and their incoherent nature frees the system from speckle (coherent noise) present in lasers.

The LED's will illuminate an acousto-optic (AO) spatial light modulator. Figure 5.3 shows the details of the optics. The AO cell will simultaneously be driven by the same side signal $n_1(t)$, so that delayed versions of that signal will be spread across the cell aperture. This aperture should be wide enough to produce sufficient delay (about 40 μ -sec) in the side signal. This use of AO cells to produce delayed signals is similar to the techniques used in the optical signal processors of [1] and [6].

The LED's produce a vector whose components are delayed versions of the side signal $n_1(t)$. The same vector is represented in the crystal aperture of the AO cell. The result of illuminating this aperture

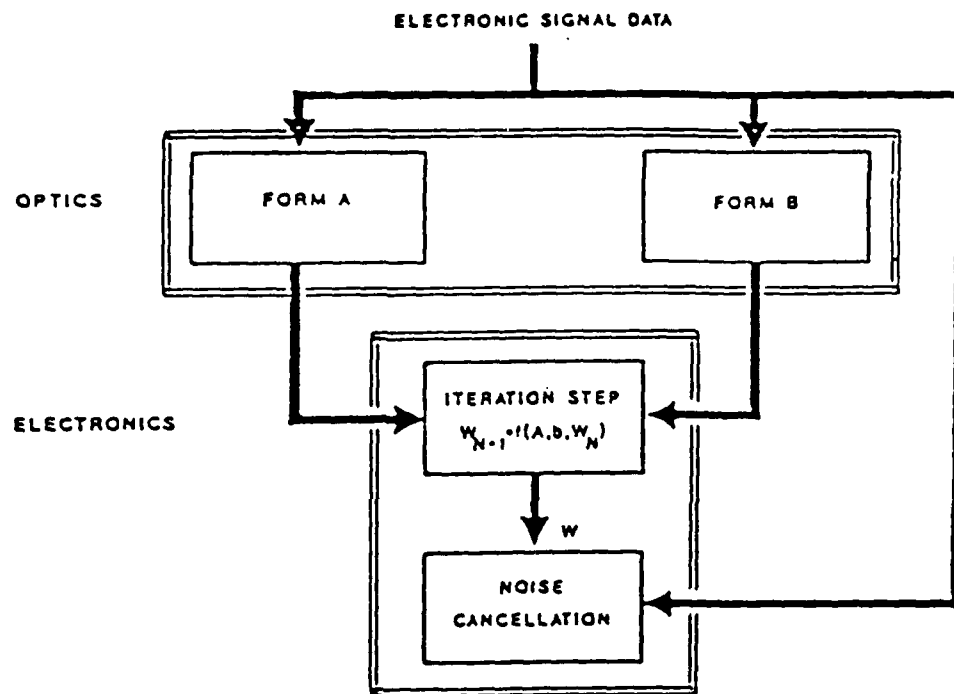


FIGURE 5.1 DIVISION OF TASKS IN HYBRID ELECTRO-OPTIC PROCESSOR

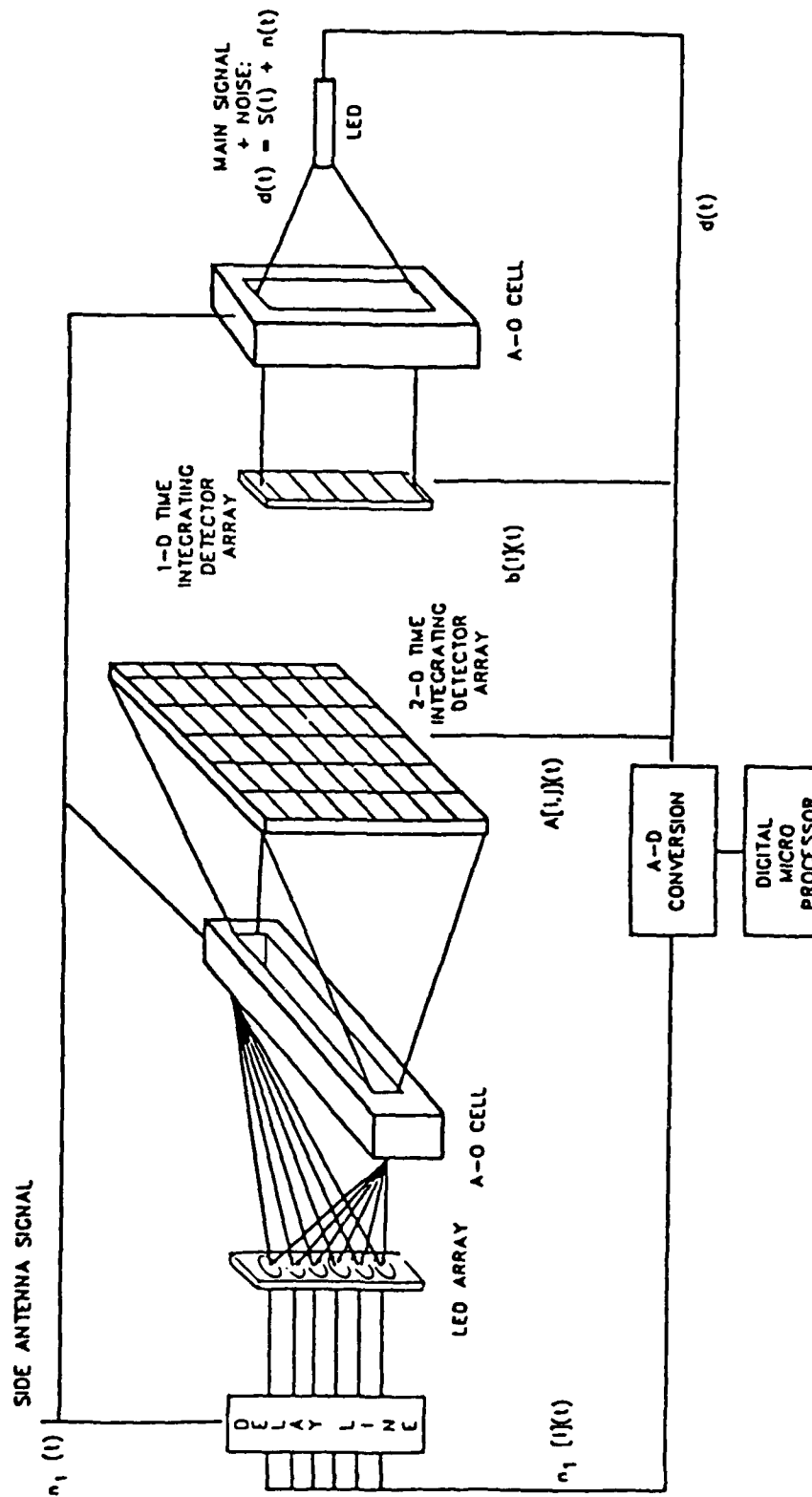
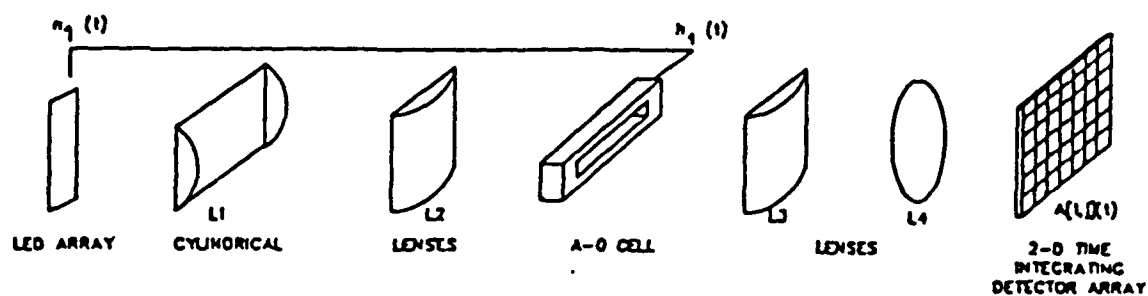
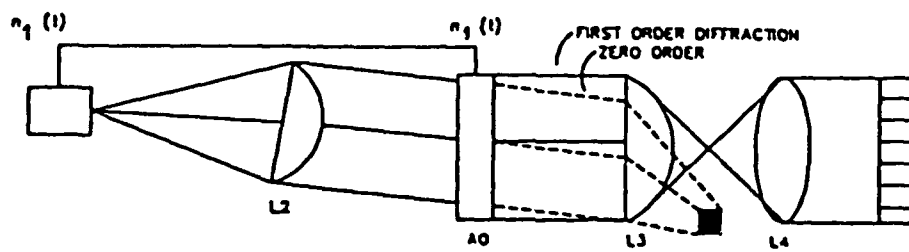


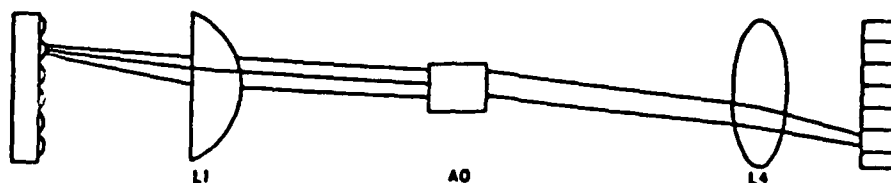
FIGURE 5.2 HYBRID SYSTEM OVERVIEW



A) OPTICAL COMPONENTS



B) TOP VIEW, SHOWING BRAGG ANGLE, AND BLOCKAGE OF ZERO ORDER DIFFRACTION



C) SIDE VIEW

FIGURE 5.3 DETAILS OF OPTICS FOR COMPUTING $A[i, j](t)$

with the LED's is the outer product of these vectors, which is a matrix. This matrix is collected and time integrated by a 2-dimensional time integrating charge coupled device (CCD) detector array. The output of the detector array is the covariance matrix A_n . A frame grabber will send the matrix data to the digital microprocessor.

To construct the vector b_n at each time step, a single LED, driven by the main signal plus noise, $s(t) + n(t)$, illuminates an AO cell which is simultaneously being driven by the side signal $n_1(t)$. The resulting modulated light represents a vector whose components are the product of $s(t) + n(t)$ with delayed versions of $n_1(t)$. This light is collected onto a one dimensional CCD time integrating detector array. The output of this detector array is b_n , which is sent to the microprocessor via an analog to digital (A/D) converter board.

The main signal plus noise, $s(t) + n(t)$, as well as the delayed versions of the side signal $n_1(t)$ from the tapped delay lines, are also sent through the A/D board to the microprocessor. The iteration step and the actual noise cancellation will be performed in the digital signal domain within the microprocessor.

Such a hybrid system should be viewed as a low cost proof of principle device that could validate this class of nonstationary iterative processes for signal processing applications. The A/D conversion would limit its usefulness as a real time processor.

5.2 All-Optical System

Figure 5.4 shows a simplified system diagram for a possible optical implementation of the nonstationary steepest descent with fixed stepsize algorithm. Only one side signal is shown, although multiple side signals could be handled with a multi-channel AO cell.

AO cells are used to produce a continuum of delayed versions of the side signals, and to form products of these delayed signals with other quantities. A lens performs spatial integration. Liquid crystal light valves (LCLV) perform time integration. The weight vector w is computed in the optic domain, and the output of the system is an optical representation of the estimated noise signal $y(t)$. This will be converted by a detector to the electronic domain where it will be recombined with the main signal plus noise, $s(t) + n(t)$, to produce the final system output, namely

$$s(t) + n(t) - y(t).$$

This signal should be close to the main signal $s(t)$.

While the nonstationary steepest descent algorithm is not the most powerful we have considered here,

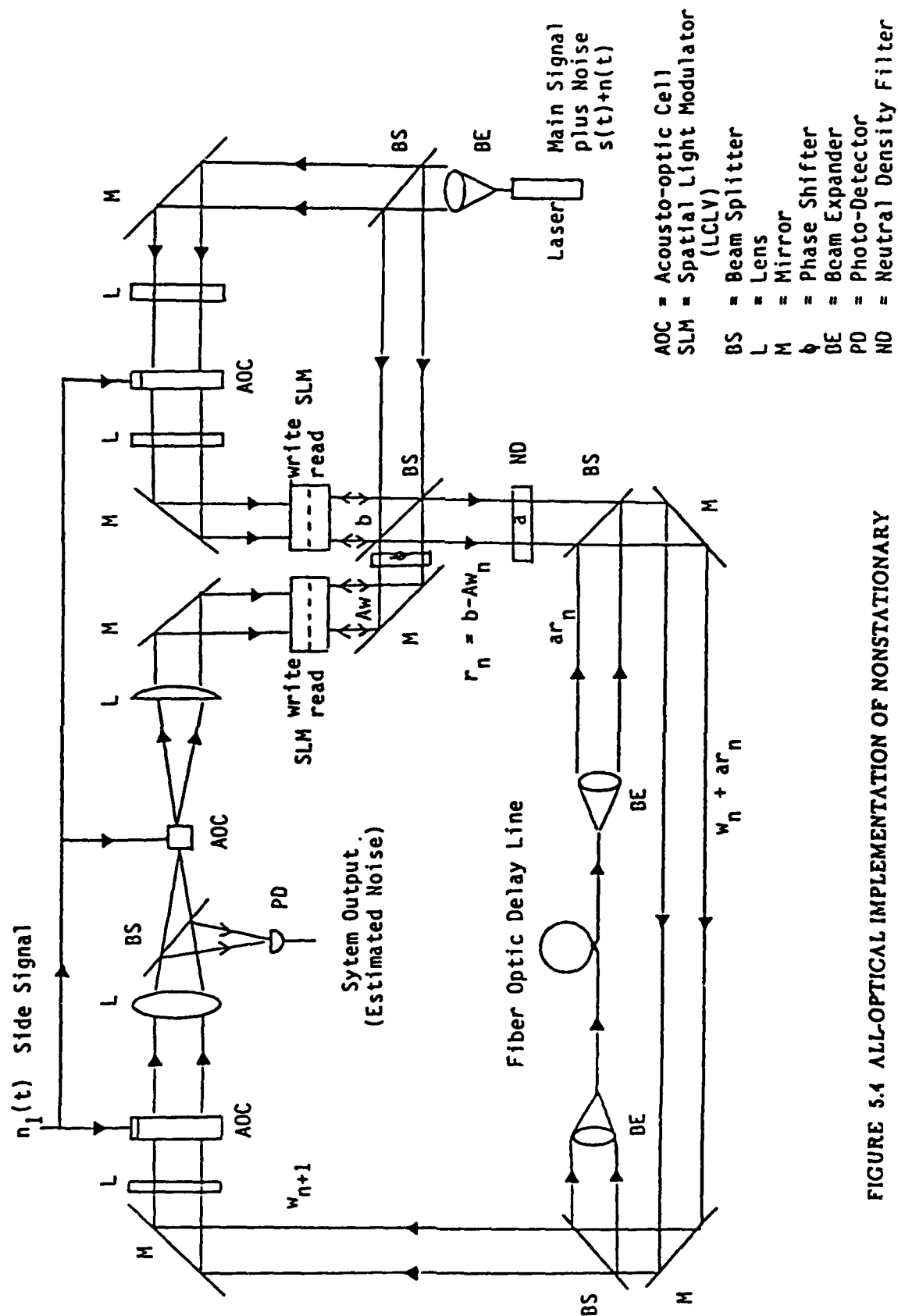


FIGURE 5.4 ALL-OPTICAL IMPLEMENTATION OF NONSTATIONARY
STEEPEST DESCENT ALGORITHM

it is the simplest to implement optically. A realization of the optical implementation for this algorithm would be an important step toward opening up this whole class of nonstationary iterative algorithms to optical implementation.

6. Concluding Remarks and Recommendations

6.1 The State of the Art

Optical signal processors have already been built which can perform adaptive noise cancellation ([1] and [6]), and optical processors implementing iterative algorithms such as steepest descent and conjugate gradient have also been built, or at least proposed ([13] and [14]). So what is new about the approach that is being presented here? The optical processors that actually take in signal data and perform adaptive noise cancellation in real time are implementing some version of the LMS algorithm, and thus suffer from the performance limitations of that algorithm. The optical processors which use true iterative algorithms such as steepest descent do so on fixed matrix data, in the form of some type of mask. Thus they are not true real time signal processors, ie., they cannot formulate the matrix problem in real time and solve it. The approach we are advancing here does propose to formulate the problem in real time and solve it with the performance advantages of iterative algorithms.

6.2 Recommendations

Nonstationary iterative algorithms can provide significant advantages over LMS for adaptive noise cancellation. Optical processing will be necessary to implement these algorithms in a real time environment because of the computational load. These algorithms are good candidates for optical implementation because they take advantage of the power of optics, rather than just mimic what is already being done electronically. The technology is here now to realize optically the simplest of these algorithms, namely nonstationary steepest descent with fixed stepsize. The means to do this was outlined in the previous section. A successful optical implementation of this algorithm would open this whole class of algorithms to optics. The numerical examples of section 3 show the potential improvement possible through the use of the conjugate gradient algorithm. This algorithm could be implemented optically if a means can be found to accomplish scalar multiplication and division in the all-optic domain. The hybrid processor discussed in the previous section provides a means of validating this entire class of algorithms for signal processing applications. If improvements can be made in A/D conversion, such a processor could find practical use.

Finally, the ultimate goal of optical computing in signal processing applications should be to produce an optical processor using integrated optics, or perhaps some three dimensional analog of integrated optics (three dimensional wave guides have already been developed). Acousto-optic cells, lenses, lasers,

delay lines, and detectors have all been fabricated in integrated optics devices, with the technology for spatial light modulators lagging somewhat behind. When integrated optics technology matures we can hope to bring optical computing techniques out of the laboratory and into the field in the form of rugged, practical devices.

References

1. V.C. Vannicola and W.A. Penn, "Acousto-Optic Adaptive Processing", GOMAC Digest of Papers, Vol. X, 1984, pp. 404-409.
2. V.C. Vannicola, W.A. Penn, and M.F. Lowry, "Recent Improvements in the Acousto-Optic Adaptive Processor", GOMAC Digest of Papers, Vol. XI, 1985, pp. 477-480.
3. S.G. Mikhlin, The Problem of the Minimum of a Quadratic Functional, Holden-Day, Inc. 1965.
4. S.T. Welstead, "Analog Algorithms for Optical Signal Processing", 20th Asilomar Conf. on Signals, Systems, and Computers, Nov. 1986, pp. 536-540.
5. B. Widrow and S.D. Stearns, Adaptive Signal Processing, Prentice Hall, Inc., 1985.
6. A. Vander Lugt, "Adaptive Optical Processor", Applied Optics, Vol. 21, No. 22, 1982, pp. 4005-4011.
7. J.B. Foley and F.M. Boland, "Comparison between Steepest Descent and LMS Algorithms in Adaptive Filters", IEEE Proc., Vol. 134, Pt. F, No. 3, 1987, pp. 283-289.
8. S.G. Mikhlin and K.L. Smolitsky, Approximate Methods for Solution of Differential and Integral Equations, Elsevier, New York, 1967.
9. J.M. Ortega and W.C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.
10. M. Hestenes, Conjugate Direction Methods in Optimization, Springer Verlag, New York, 1980.
11. S.T. Welstead, "Iterative Algorithms for an Optical Signal Processor", 30th Midwest Symposium on Circuits and Systems, Elsevier, 1988, pp. 490-493.
12. S.T. Welstead, "Real Time Iterative Algorithms for Optical Signal Processing", SPIE Vol. 827, Real Time Signal Processing X, 1987, pp. 137-144.
13. A.K. Ghosh, D. Casasent, and C.P. Neuman, "Performance of Direct and Iterative Algorithms on an Optical Systolic Processor", Applied Optics, Vol. 24, No. 22, 1985, pp. 3883-3892.
14. A.K. Ghosh, "Realization of Lanczos and Conjugate Gradient Algorithms on Optical Linear Algebra Processors", SPIE Vol. 827, Real Time Signal Processing X, 1987, pp. 208-215.

Publications and Presentations

Publications supported by this contract:

S. T. Welstead, "Real Time Iterative Algorithms for Optical Signal Processing", SPIE VOL. 827, Real Time Signal Processing X, 1987, pp. 137-144.

S. T. Welstead, "Iterative Algorithms for an Optical Signal Processor", 30th Midwest Symposium on Circuits and Systems, Elsevier, 1988, pp. 490-493.

Presentations supported by this contract:

"Real Time Iterative Algorithms for Optical Signal Processing", SPIE, San Diego, August, 1987.

"Iterative Algorithms for an Optical Signal Processor", 30th Midwest Symposium on Circuits and Systems, Syracuse, August, 1987.

"Real Time Iterative Algorithms for Optical Signal Processing", University of Alabama in Huntsville Department of Mathematics and Statistics Colloquium, November, 1987.

Future presentation on work performed during this contract:

"Iterative Algorithms for Real Time Signal Processing", SIAM's Third Conference on Applied Linear Algebra, Madison, WI, May 23-26, 1988.

Appendix 1

Preliminary Study of an Optical Implementation of the Conjugate Gradient Algorithm

1986 USAF-UES Summer Faculty Research Program/
Graduate Student Summer Support Program

Sponsored by the
Air Force Office of Scientific Research
Conducted by the
Universal Energy Systems, Inc.
Final Report

Preliminary Study of an Optical Implementation
of the Conjugate Gradient Algorithm

Prepared by: Stephen T. Welstead
Academic Rank: Assistant Professor
Department and Mathematics and Statistics
University: University of Alabama in Huntsville
Research Location: Rome Air Development Center, OCTS,
Griffiss Air Force Base, New York
USAF Research: Dr. Vincent Vannicola
Date: August 15, 1986
Contract No.: F49620-85-C-0013

Preliminary Study of an Optical Implementation
of the Conjugate Gradient Algorithm

by

Stephen T. Welstead

ABSTRACT

An analysis was done on an acousto-optic signal processor in order to make recommendations about possible improvements in the algorithm being used. The optical nature of the processor necessitates looking at data in an analog, rather than digital, fashion. The minimization problem, which the processor is solving, is formulated in an analog way. This leads to an operator equation in Hilbert space, rather than the usual matrix equation. Operator theoretic versions of steepest descent and conjugate gradient algorithms are discussed. Block diagrams are given for these algorithms, along with recommendations for possible optical implementations.

I. Introduction

My Ph.D. dissertation topic at Purdue University concerned the numerical solution of integral equations. After several years of working in that area, I decided to branch out into other areas, and began looking at optical signal processing and image processing. Through the Center for Applied Optics at the University of Alabama in Huntsville I was able to do funded research in optics in 1985.

The research problem I selected at the Rome Air Development Center, Griffiss AFB, NY, concerns the investigation of different mathematical algorithms for implementation on an acousto-optic signal processor. Because of the analog nature of the optics involved, the problem to be solved turns out to be an integral equation. The combination of optics and integral equations makes this problem particularly well suited to my background.

II. Objectives of the Research Effort

The objective of the research effort is to improve the performance of an acousto-optic signal processor (already in experimental operation) by implementing a more efficient mathematical algorithm. The system in operation now uses a Howells-Applebaum least mean square (LMS) algorithm. It was felt that performance could be improved if a more powerful algorithm, such as the conjugate gradient algorithm, were implemented. My individual objectives were:

1. Familiarize myself with the acousto-optic processor now in operation in order to fully understand its implementation of the LMS algorithm.
2. Formulate the mathematical problem to be solved, keeping in mind the special nature of the optical processing involved.
3. Study the conjugate gradient and related algorithms and investigate the feasibility of implementing these algorithms in an optical system similar to the one now in operation.
4. Construct a block diagram for the conjugate gradient algorithm and make recommendations about possible optical

implementations.

III. Analog Formulation of the Minimization Problem for the Optical System

We consider the signal processing problem of cancelling noise from a main signal. The receiving configuration consists of a main antenna and N omni-directional side antennas. An acousto-optic processor for such a system has been proposed and implemented by Vannicola and Penn [VP1,VP2]. A similar system has been considered by Vander Lugt [V1].

We denote by $n_1(t), n_2(t), \dots, n_N(t)$ the signals received at the side antennas at time t , and by $s(t) + n(t)$ the main signal plus noise received at the main antenna. Each side channel signal $n_i(t)$ is input through an acousto-optic device which produces a continuum of delayed signals $n_i(t-x)$, where the delay x ranges from 0 to a value R which depends on the acousto-optic device (R is typically in the range 5-50 microseconds). We call x a 'spatial' variable here, since it represents position across the acousto-optic device, although it can also be thought of as another time variable.

Our problem is to form a weighted combination of the delayed secondary signals $n_i(t-x)$ in such a way that the result is a good estimate of the noise $n(t)$ received at the main channel. The continuous nature of the delays necessitates that we look at this problem in an analog way, rather than the usual discrete formulation involving matrices and vectors. Thus, we define a Hilbert space H consisting of the set of complex vector valued functions $\underline{h}(x) = (h_1(x), \dots, h_N(x))$ defined on the real interval $[0, R]$ with inner product

$$(\underline{h}, \underline{g}) = \sum_{i=1}^N \int_0^R h_i(x) \overline{g_i(x)} dx$$

where for a complex variable z , \bar{z} denotes its complex conjugate.

Define the functions of two variables $f_i(x,t) = n_i(t-x)$, $i = 1, \dots, N$, and let $\underline{f}(x,t)$ be the vector valued function whose i^{th} component is $f_i(x,t)$. Our problem, then, is to determine a vector valued weight function $\underline{w}(x) = (w_1(x), \dots, w_N(x))$ so that the scalar function

$$y(t) = (\underline{f}(\cdot, t), \underline{w}) \quad (3.1)$$

is a good approximation of the noise $n(t)$.

The output of the system is the "error" signal $e(t)$, which is the main signal plus noise minus the estimated noise:

$$e(t) = s(t) + n(t) - y(t).$$

The quantity we wish to minimize is

$$E(|e(t)|^2) = \int_0^\infty |e(t)|^2 dt. \quad (3.2)$$

E can be thought of as an expected value over time, although for the purposes of our minimization problem we have omitted any reference to a probability density function. (One can also think of (3.2) as an "energy" integral.) Setting $d(t) = s(t) + n(t)$, we find

$$\begin{aligned} |e(t)|^2 &= e(t)\overline{e(t)} \\ &= (d(t) - y(t))(\overline{d(t)} - \overline{y(t)}) \\ &= |d(t)|^2 - y(t)\overline{d(t)} - d(t)\overline{y(t)} + |y(t)|^2. \end{aligned}$$

Then

$$\begin{aligned} E(y(t)\overline{d(t)}) &= \int_0^\infty y(t)\overline{d(t)} dt \\ &= \int_0^\infty \overline{d(t)} \left\{ \sum_{j=1}^N \int_0^R f_j(x,t) \overline{w_j(x)} dx \right\} dt \\ &= \sum_{j=1}^N \int_0^R \left\{ \int_0^\infty f_j(x,t) \overline{d(t)} dt \right\} \overline{w_j(x)} dx \end{aligned}$$

$$= (\underline{b}, \underline{w})$$

where

$$\underline{b}(x) = \begin{pmatrix} \int_0^1 f_1(x, t) \overline{d(t)} dt \\ \vdots \\ \int_0^N f_N(x, t) \overline{d(t)} dt \end{pmatrix}$$

Similarly,

$$\begin{aligned} E(\overline{y(t)} d(t)) &= \overline{E(y(t) d(t))} \\ &= \overline{(\underline{b}, \underline{w})} \\ &= (\underline{w}, \underline{b}). \end{aligned}$$

Also, using (3.1), we find

$$\begin{aligned} E(|y(t)|^2) &= \int_0^\infty y(t) \overline{y(t)} dt \\ &= \int_0^\infty \left\{ \sum_{j=1}^N \int_0^R f_j(x, t) \overline{w_j(x)} dx \right\} \overline{(f(\cdot, t), \underline{w})} dt \\ &= \sum_{j=1}^N \int_0^R \overline{w_j(x)} \left\{ \int_0^\infty f_j(x, t) \overline{(f(\cdot, t), \underline{w})} dt \right\} dx \\ &= (A \underline{w}, \underline{w}), \end{aligned}$$

where A is an operator mapping H to H, defined by

$$\underline{Aw}(x) = \begin{pmatrix} \int_0^\infty f_1(x,t) \overline{f(\cdot,t),w} dt \\ \cdot \\ \cdot \\ \cdot \\ \int_0^\infty f_N(x,t) \overline{f(\cdot,t),w} dt \end{pmatrix}$$

The j^{th} component of \underline{Aw} can be rewritten as

$$\begin{aligned} \int_0^\infty f_j(x,t) \sum_{k=1}^N \int_0^R \overline{f_k(s,t)} w_k(s) ds dt &= \\ &= \sum_{k=1}^N \int_0^R w_k(s) \int_0^\infty f_j(x,t) \overline{f_k(s,t)} dt ds \\ &= (\underline{w}, \underline{A_j}(x, \cdot)), \end{aligned}$$

where $\underline{A_j}(x,s)$ is a vector valued function of two spatial variables whose k^{th} component is given by

$$A_{jk}(x,s) = \int_0^\infty f_k(s,t) \overline{f_j(x,t)} dt$$

for $k = 1, \dots, N$.

The operator A can be thought of as an analog "outer product". It corresponds to the cross-correlation (or covariance) matrix in the usual discrete formulation of the problem.

It is straightforward to show that for functions $\underline{w}, \underline{u}$ in H we have

$$(\underline{Aw}, \underline{u}) = (\underline{w}, \underline{Au})$$

so that A is a self adjoint operator on H . Also, one can show

$$(\underline{A}\underline{u}, \underline{u}) = \int_0^\infty |(\underline{u}, \underline{f}(\cdot, t))|^2 dt. \quad (3.3)$$

The expression on the right side of (3.3) is > 0 for $\underline{u} \neq 0$, thus A is a positive operator on H .

Our minimization problem can thus be reformulated as the problem of minimizing the functional F defined by

$$F(\underline{w}) = E(|\underline{d}(t)|^2) - (\underline{b}, \underline{w}) - (\underline{w}, \underline{b}) + (\underline{A}\underline{w}, \underline{w}) \quad (3.4)$$

The right side of (3.4) is just $E(|e(t)|^2)$ (see (3.2)). The quantity $E(|\underline{d}(t)|^2)$ is independent of \underline{w} and is of no consequence in the minimization problem. Since A is a positive operator, the problem of minimizing (3.4) is equivalent to solving the operator equation

$$\underline{A}\underline{w} = \underline{b} \quad (3.5)$$

([M, Theorem 2.1]). Equation (3.5) can also be written as a coupled system of integral equations:

$$\sum_{k=1}^N \int_0^R w_k(s) \overline{A_{jk}(x, s)} ds = b_j(x), \quad j = 1, \dots, N \quad (3.6)$$

IV. The Existing Architecture

Before examining any new algorithms for the solution of (3.5), let us first look at what the existing optical system (reported in [VP1, VP2]) is doing. Figure 1 is a simplified diagram of this system, showing one side channel only.

This architecture is implementing the LMS algorithm, which is an approximate version of the method of steepest descent. Assume, for the moment, that we are receiving signal samples at discrete time intervals $t = i\Delta t$ for some fixed Δt . Consider the following iterative scheme for determining the weight function $w(x)$ (one side channel only):

$$w_{i+1}(x) = w_i(x) + a_i p_i(x). \quad (4.1)$$

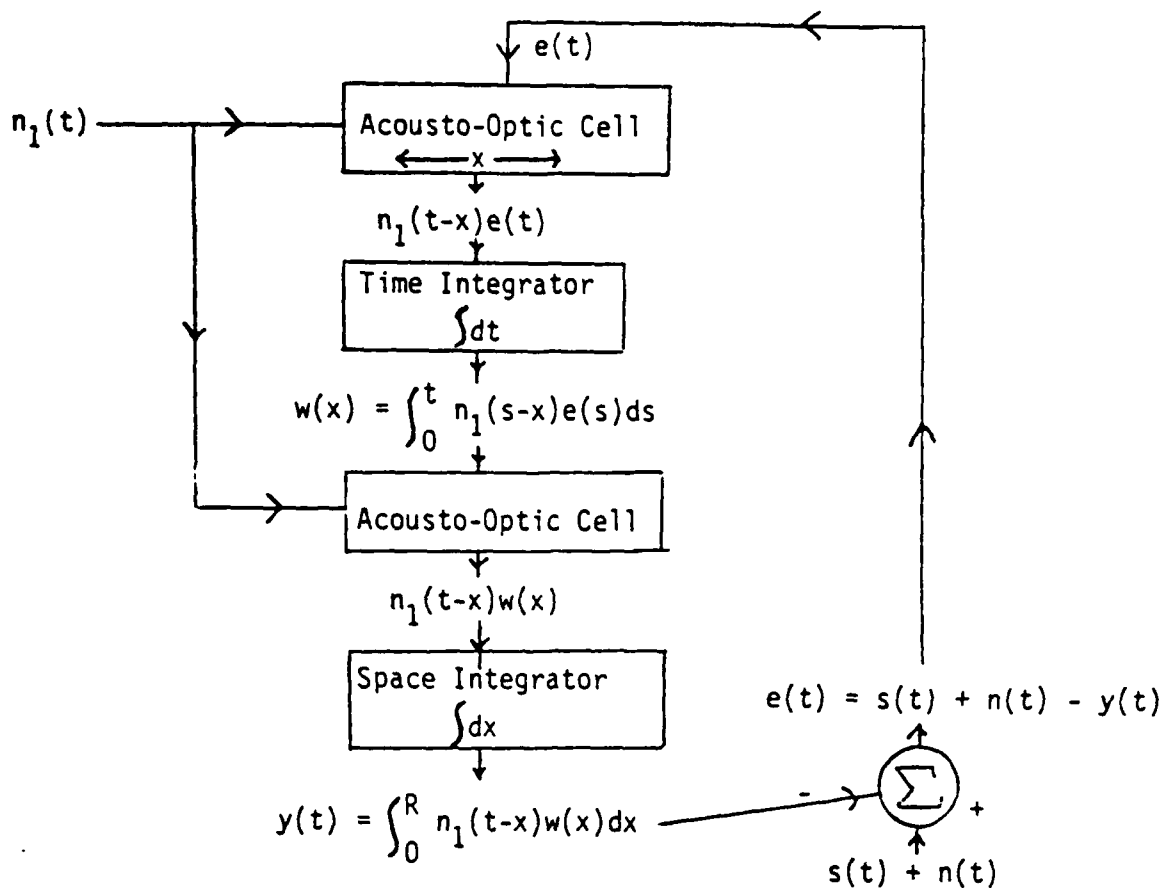


Figure 1
BLOCK DIAGRAM FOR CURRENT SYSTEM

The functions $p_i(x)$ are direction 'vectors', a_i are scalars, and $w_i(x)$ is the i^{th} iterative approximation of $w(x)$. The method of steepest descent uses, for $p_i(x)$, the negative gradient of the functional F , defined by (3.4), evaluated at $w_i(x)$. This gradient is $-2r_i(x)$ (cf., [L]) where

$$r_i(x) = b(x) - Aw_i(x) \quad (4.2)$$

is the i^{th} residual.

The LMS method uses the gradient of $|e_i|^2$, instead of the gradient of $F(w) = E(|e_i|^2)$, where $e_i = e(i\Delta t)$. This gradient can be computed as $-2e_i n_1(i\Delta t - x)$. The algorithm thus becomes

$$w_{i+1}(x) = w_i(x) + a_i e_i n_1(i\Delta t - x) \quad (4.3)$$

where we have absorbed the factor 2 as part of a_i . Notice that now we do not have to compute the operator A , as is necessary in determining the residual defined by (4.2).

The iterative scheme given by (4.3) can be easily solved by observing that

$$\sum_{i=0}^{K-1} w_{i+1}(x) - w_i(x) = w_K(x) - w_0(x).$$

Taking $w_0(x) = 0$, we obtain from (4.3)

$$w_K(x) = \sum_{i=0}^{K-1} a_i e_i n_1(i\Delta t - x). \quad (4.4)$$

We now make the assumption that the step size a_i (or "beam steering signal" [MM]) has been incorporated into the signal n_1 . Letting $\Delta t \rightarrow 0$, we get the analog version of (4.4), namely

$$w(x) = \int_0^t e(s) n_1(s-x) ds. \quad (4.5)$$

This is the first integral which appears in Figure 1. It can also be interpreted as a correlation of n_1 with the "error" signal $e(t)$.

The advantage of this implementation is its simplicity. There is no iteration loop, rather, the iteration scheme has been solved directly, and an expression for the solution implemented. Also, the problem of computing the operator A has been avoided completely.

The disadvantage is that it may not produce an accurate solution for $w(x)$. The method of steepest descent typically can have very slow convergence, and one would expect this LMS method to be even slower.

V. The Conjugate Gradient Algorithm

We now consider another iterative method for solving equation (3.5). We return to the iteration equation (4.1), which we now write for the case of multiple side channels, so that \underline{w}_i

(x) and $p_i(x)$ are elements of the Hilbert space H :

$$w_{i+1}(x) = w_i(x) + a_i p_i(x).$$

We now choose the direction vectors $p_i(x)$ to be a set of linearly independent, A -orthogonal vectors, ie., the $p_i(x)$ are such that

$$(p_i, A p_j) = 0 \quad \text{for } i \neq j. \quad (5.1)$$

The scalar a_i is chosen at each step of the iteration process to minimize the value of $F(w_i)$. The iteration method is then said to be a conjugate directions method (the use of the word 'conjugate' here comes from the fact that vectors satisfying (5.1) are said to be A -conjugate).

If one chooses as the vectors $p_i(x)$ the A -orthogonalized residuals $r_i(x) = b(x) - A w_i(x)$ then one obtains the conjugate gradient method. This method can be summarized by the following iteration scheme (cf., [L]):

$$\begin{aligned} w_{i+1}(x) &= w_i(x) + a_i p_i(x) \\ p_{i+1}(x) &= r_{i+1}(x) - c_i p_i(x) \\ a_i &= (r_i, p_i) / (p_i, A p_i) \\ c_i &= (r_{i+1}, A p_i) / (p_i, A p_i) \\ r_i(x) &= b(x) - A w_i(x). \end{aligned} \quad (5.2)$$

The value of a_i comes from minimizing $F(w_i)$, and the value of c_i comes from A -orthogonalizing the vectors $r_i(x)$. (It is a nontrivial property of this method that one need only A -orthogonalize $p_{i+1}(x)$ with respect to $p_i(x)$, and not all the preceding $p_j(x)$'s, to obtain a complete A -orthogonal set (cf. [L]).)

What is the motivation for considering conjugate direction methods? One reason is the following fact. Suppose we have some weight value $w_0(x)$ and we compute a new value $w_1(x)$ from

$$\underline{w}_1(x) = \underline{w}_0(x) + a_0 \underline{p}_0(x)$$

where $\underline{p}_0(x)$ is any nonzero direction vector and a_0 is chosen to minimize $F(\underline{w}_1)$ (so a_0 is given by the expression in (5.2) with $i=0$). Now suppose $\underline{w}^*(x)$ is the true solution of (3.5). Then the correct direction to go in, in order to reach exactly \underline{w}^* on the next step, is always going to be A -orthogonal to the previous direction vector used (in this case, $\underline{p}_0(x)$). To see this, note that the direction from \underline{w}_1 to \underline{w}^* is $\underline{w}^* - \underline{w}_1$ and

$$\begin{aligned} (\underline{p}_0, A(\underline{w}^* - \underline{w}_1)) &= (\underline{p}_0, \underline{b} - A\underline{w}_1) \\ &= (\underline{p}_0, \underline{b} - A(\underline{w}_0 + a_0 \underline{p}_0)) \\ &= (\underline{p}_0, \underline{b} - A\underline{w}_0) - a_0 (\underline{p}_0, A\underline{p}_0) \\ &\quad (a_0 \text{ is real}) \\ &= (\underline{p}_0, \underline{r}_0) - a_0 (\underline{p}_0, A\underline{p}_0) \\ &= 0 \quad \text{from (5.2).} \end{aligned}$$

In the discrete case, when A is a finite dimensional matrix, there are only finitely many directions which are A -orthogonal to a given vector. Conjugate direction methods search through this finite list until exactly the right direction vector is found. They are thus guaranteed to converge to the exact solution (ignoring roundoff errors) in a finite number of steps. In contrast, if the method of steepest descent does not obtain the exact solution in one step, then it will always take infinitely many steps to reach the exact solution ([LS]).

We are not dealing with a finite dimensional matrix, but rather with an "infinite dimensional" operator A , so the finite step advantage mentioned above is not, in general, applicable to our situation (there are, however, cases when finite convergence is attained for an operator A (cf. [LS])). However, the conjugate gradient method will always converge more rapidly than the method of steepest descent (see [D] for estimates on the rate of convergence).

VI. Block Diagrams

In this section, block diagrams for two iterative methods are presented. These diagrams are constructed with optical implementation in mind (eg., there is no storage of data or previously computed results). Boxes labeled "compute A", "compute b", etc., are indicated in detail in separate diagrams.

Figure 2 shows a diagram for the method of steepest descent. This method is included here because it is simpler to implement than the conjugate gradient algorithm, yet it contains most of the computational difficulties (computing A, b, inner products, and inverting scalars). If this method can be implemented optically, then it would be relatively straightforward to modify the resulting system for the conjugate gradient algorithm.

Figure 6 shows the diagram for the conjugate gradient algorithm. As one can see, it contains all of the computations required by steepest descent, plus additional computations required for obtaining the vectors $\underline{p}_i(x)$. Since we have not assumed the possibility of storing previously computed results, we must compute both \underline{r}_i and \underline{r}_{i+1} in each iteration loop. Each of these residual computations requires the computation of A and b (see Figures 3-4).

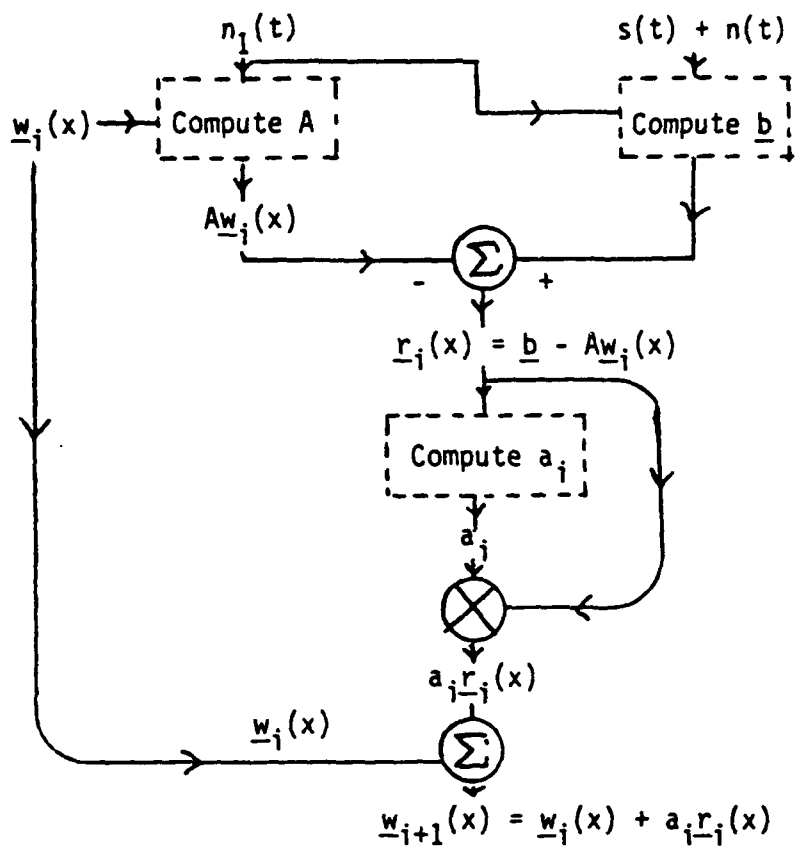


Figure 2. THE METHOD OF STEEPEST DESCENT

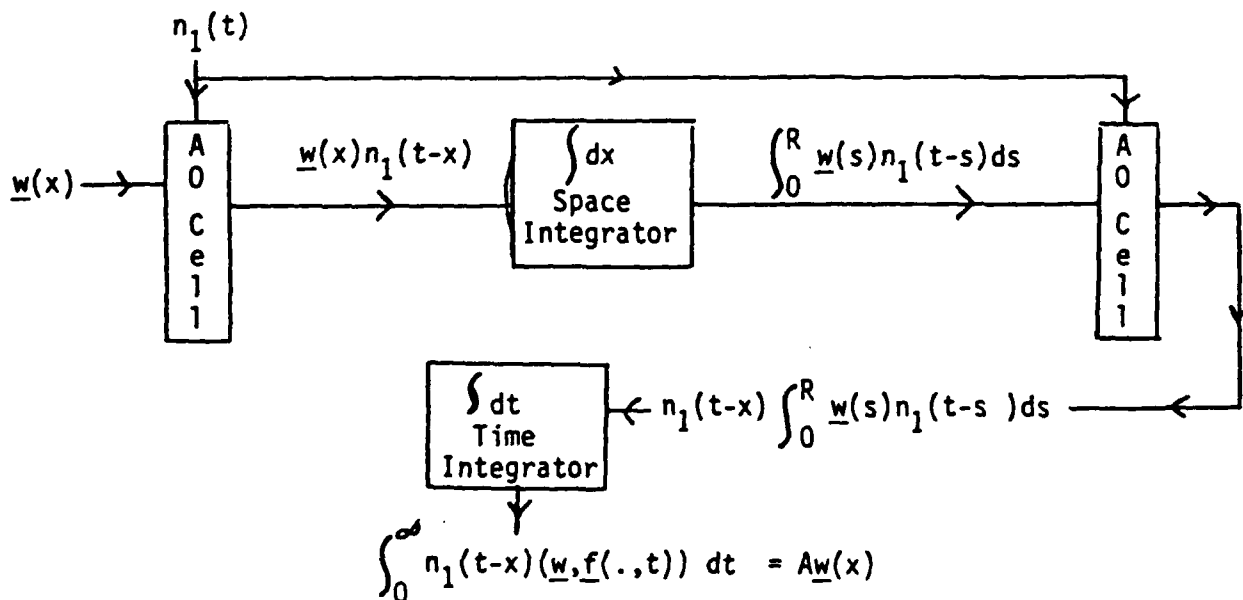


Figure 3. COMPUTING $A \underline{w}(x)$

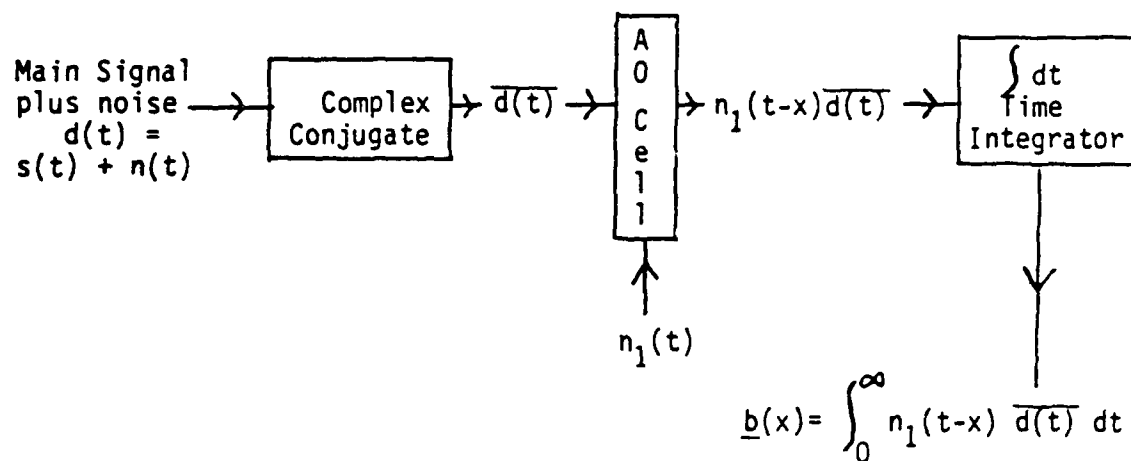


Figure 4. COMPUTING $\underline{b}(x)$

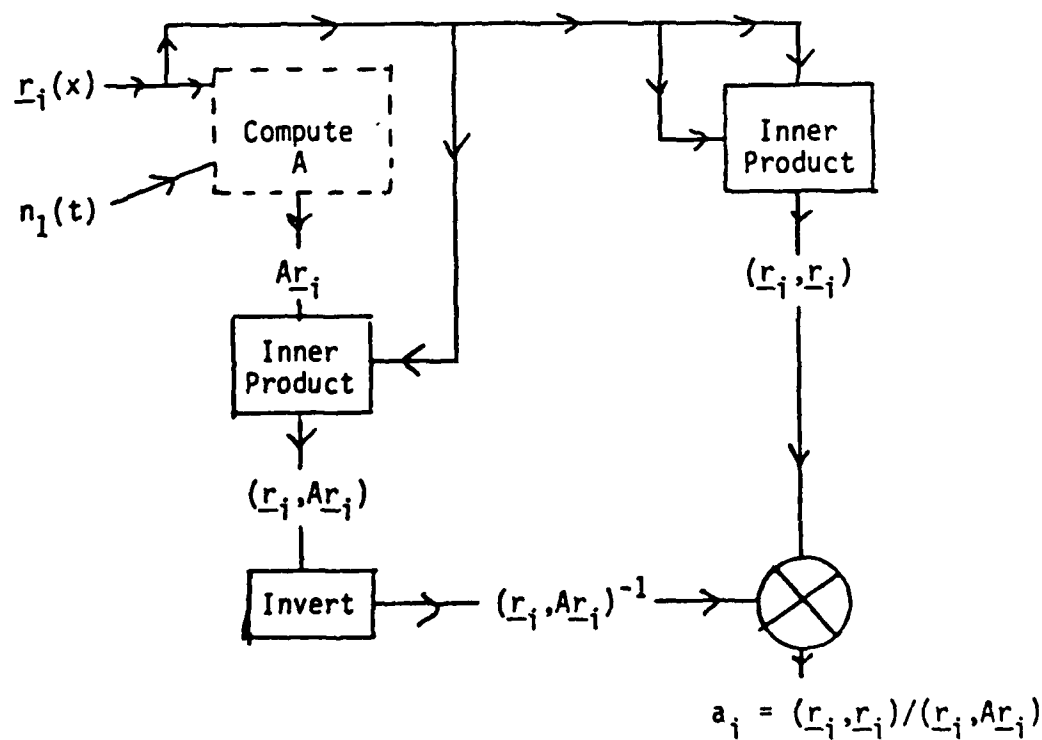
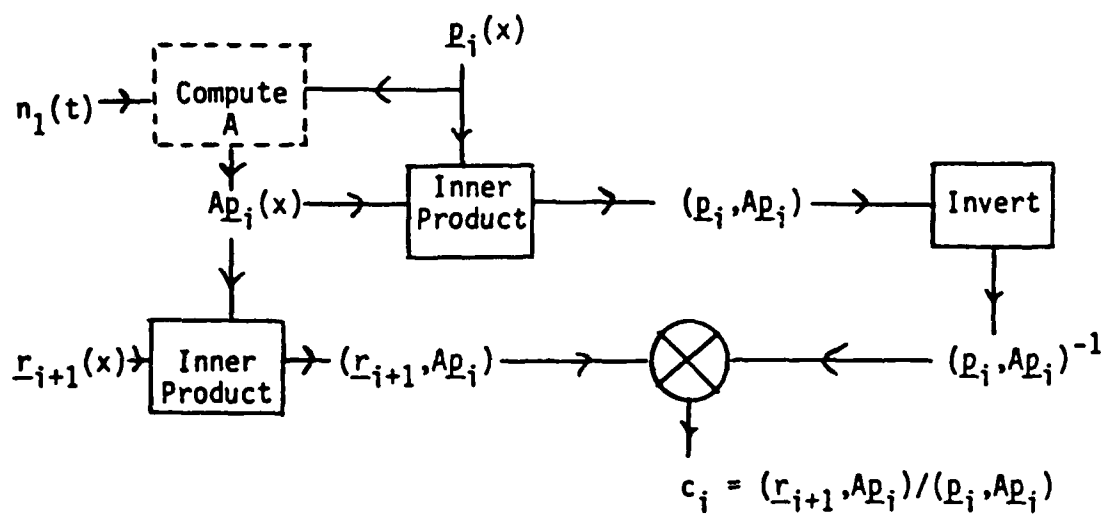
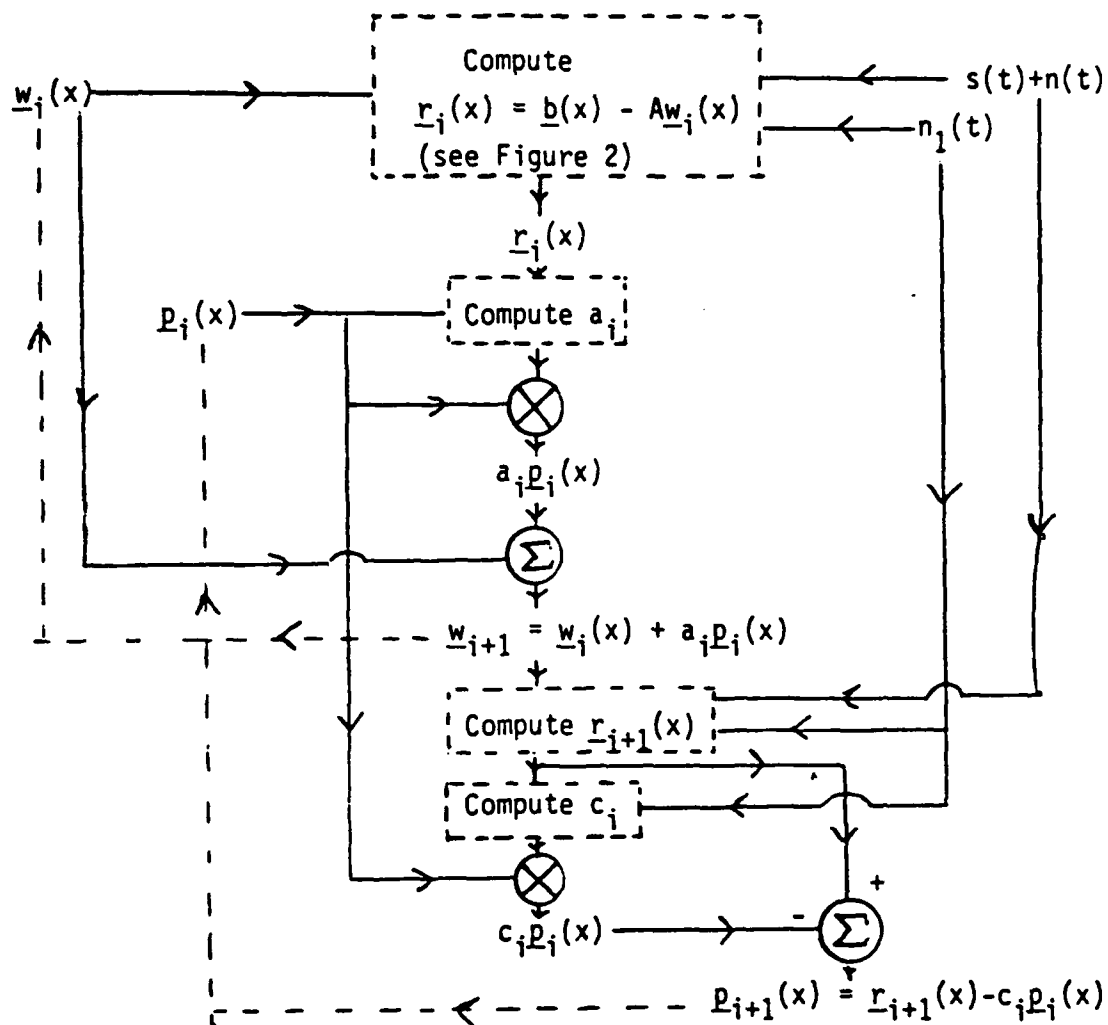


Figure 5. COMPUTING a_i



VII. Recommendations

In order to achieve an optical implementation of either of the iterative methods discussed in the previous section, we must first be able to implement the individual operations shown in the block diagrams. The major operations are:

1. Computation of the operator A (Figure 3). This requires two acousto-optic cells and space and time integration. In the current system, space integration is done with a lens, and time integration with a liquid crystal light valve (LCLV). In fact, all the operations needed to compute A are done in the current system, so this should present no problem. Since A essentially represents an "outer product", reference [A], which discusses optical computation of outer products, may be helpful.
2. Computation of b (Figure 4). This requires another acousto-optic cell, and a time integration (LCLV).
3. Inner Products. Each inner product requires complex conjugation, pointwise multiplication, space integration, and summation. Also, both vectors will be represented as light, so an acousto-optic cell (which has one electronic input) may not be appropriate. An efficient optical means will have to be found to compute these inner products.
4. Inverting Scalars. This may be the hardest operation to implement optically. It may require its own iterative loop.

The iterative loops involved in both the steepest descent and conjugate gradient algorithms are a major departure from the existing optical system. There are two alternatives to approaching the implementation of these loops, both having to do with the idea that the weights are supposed to be slowly varying with time.

The first approach would be to consider taking in data in blocks, rather than continuously, and doing a set number of iterations on each block of data to determine the weights. The value of the weights would be updated as each block of data comes in. This would be a true implementation of the algorithms as outlined above.

A second approach, which requires further analysis but would be easier to implement, would be to do just one iteration as part of the existing loop. Since the weights are assumed "constant" in time, this would have the effect of many iterations as new data is continuously brought in and sent through the system. Also, it would be "adaptive", just as the present system is, in that changes in the data should eventually be reflected in changes in the weights. This implementation would not too different from the existing system, but further analysis is needed to determine if the algorithms are still valid when the iterations are slowly varying in time.

My recommendation is that further analysis of the second approach mentioned above be carried out. It should be compared to the first approach, ie., the standard implementation of the algorithms. A computer simulation study comparing both should be done. If an optical implementation seems feasible, it should be carried out for the steepest descent method first, since most of the computational difficulties are encountered there. The conjugate gradient algorithm can be implemented as a straightforward modification of steepest descent.

References

- [A] Athale, R. A., and Lee, J. N., "Optical Processing Using Outer Product Concepts", Proc. IEEE, 72, No. 7, 1984, p. 931.
- [D] Daniel, J. W., "The Conjugate Gradient Method for Linear and Nonlinear Operator Equations", SIAM J. Numer. Anal., Vol. 4, No. 1, 1967, p. 18.
- [L] Luenberger, D. G., Optimization by Vector Space Methods, Wiley, 1969.
- [LS] Lasdon, L. S., Mitter, S. K., Waren, A. D., "The Conjugate Gradient Method for Optimal Control Problems", IEEE Trans. on Automat. Control, AC-12, No. 2, 1967, p. 132.
- [M] Mikhlin, S. G., The Problem of the Minimum of a Quadratic Functional, Holden-Day, Inc., 1965.
- [MM] Monzingo, R. A., and Miller, T. W., Introduction to Adaptive Arrays, Wiley, 1980.
- [V1] Vander Lugt, A., "Adaptive Optical Processor", Applied Optics, Vol. 21, No. 22, p. 4005.
- [VP1] Vannicola, V. C., and Penn, W. A., "Acousto-Optic Adaptive Processing", GOMAC Digest of Papers, Vol X, 1984.
- [VP2] _____, and Lowry, M. F., "Recent Improvements in the Acousto-Optic Adaptive Processor", GOMAC Digest of Papers, Vol. XI, 1985.